

DEVICE AND METHOD FOR EXTRACTING TITLE FROM DOCUMENT IMAGE

Publication number: JP9134406

Publication date: 1997-05-20

Inventor: KATSUYAMA YUTAKA; NAOI SATOSHI

Applicant: FUJITSU LTD

Classification:

- international: **G06K9/20; G06T11/60; G06K9/20; G06T11/60;** (IPC1-7): G06K9/20; G06T11/60

- European: G06K9/20L4

Application number: JP19950341983 19951228

Priority number(s): JP19950341983 19951228; JP19950229508 19950906

Also published as:



EP0762730 (A2)

US6035061 (A1)

EP0762730 (A3)

CN1495660 (A)

EP0762730 (B1)

more >>

[Report a data error here](#)

Abstract of **JP9134406**

PROBLEM TO BE SOLVED: To easily extract a title part from a document image obtained by converting a document into image data. **SOLUTION:** In the title extracting device, black image elements in a document are scanned, a rectangular area circumscribed to an area connecting the black image elements is extracted as a character rectangle, plural adjacent character rectangles are integrated, and a rectangular area circumscribed to the character rectangle is extracted as a character string rectangle. Then points indicating the likeliness of a title are calculated based upon attributes such as the underline attribute and framing attribute of each character string rectangle and a ruled line attribute and a character string rectangle acquiring high points is extracted as a title rectangle. In the case of a table format document, a title rectangle can be extracted also from the inside of a table. Characters in the extracted title rectangle are used as a keyword for the document image after its recognizing processing.

Data supplied from the **esp@cenet** database - Worldwide

Family list**17** family members for: **JP9134406**

Derived from 10 applications

[Back to JP913](#)

- 1 Title extracting apparatus for extracting title from document image and method thereof**
Inventor: KATSUYAMA YUTAKA (JP); NAOI SATOSHI
Applicant: FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+3)
Publication info: **CN1094224C** - 2002-11-13
CN1153955 A - 1997-07-09
- 2 Title extracting device and its method for extracting title from file images**
Inventor: HIROSHI KATSUYAMA (JP); AKI NAOI **Applicant:** FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+6)
Publication info: **CN1220162C** - 2005-09-21
CN1365078 A - 2002-08-21
- 3 Title extracting device and its method for extracting title from file images**
Inventor: HIROSHI KATSUYAMA (JP); AKI NAOI **Applicant:** FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+6)
Publication info: **CN1220163C** - 2005-09-21
CN1365079 A - 2002-08-21
- 4 Header extracting device and method for extracting header from file image**
Inventor: YUTAKA KATSUYAMA (JP); SO NAOI (JP) **Applicant:** FUJITSU LTD (JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+6)
Publication info: **CN1269068C** - 2006-08-09
CN1495660 A - 2004-05-12
- 5 Title extracting device and its method for extracting title from file images**
Inventor: HIROSHI KATSUYAMA (JP); AKI NAOI **Applicant:** FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+6)
Publication info: **CN1365080 A** - 2002-08-21
- 6 Method and apparatus for extracting a title from a scanned document**
Inventor: KATSUYAMA YUTAKA (JP); NAOI SATOSHI
Applicant: FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+2)
Publication info: **DE69624433D D1** - 2002-11-28
- 7 Method and apparatus for extracting a title from a scanned document**
Inventor: KATSUYAMA YUTAKA (JP); NAOI SATOSHI
Applicant: FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+2)
Publication info: **DE69624433T T2** - 2003-03-06
- 8 Method and apparatus for extracting a title from a scanned document**
Inventor: KATSUYAMA YUTAKA (JP); NAOI SATOSHI
Applicant: FUJITSU LTD (JP)
(JP)
EC: G06K9/20L4 **IPC:** G06K9/20; G06T11/60; G06K9/20 (+2)
Publication info: **EP0762730 A2** - 1997-03-12
EP0762730 A3 - 1998-01-28
EP0762730 B1 - 2002-10-23

**9 DEVICE AND METHOD FOR EXTRACTING TITLE FROM DOCUMENT
IMAGE**

Inventor: KATSUYAMA YUTAKA; NAOI SATOSHI **Applicant:** FUJITSU LTD

EC: G06K9/20L4

IPC: G06K9/20; G06T11/60; G06K9/20 (+3)

Publication info: JP3425834B2 B2 - 2003-07-14

JP9134406 A - 1997-05-20

**10 Title extracting apparatus for extracting title from document image and
method thereof**

Inventor: KATSUYAMA YUTAKA (JP); NAOI SATOSHI
(JP) **Applicant:** FUJITSU LTD (JP)

EC: G06K9/20L4

IPC: G06K9/20; G06T11/60; G06K9/20 (+2)

Publication info: US6035061 A - 2000-03-07

Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-134406

(43) 公開日 平成9年(1997)5月20日

(51) Int.Cl. ⁶	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 K 9/20	3 4 0		G 0 6 K 9/20	3 4 0 J
G 0 6 T 11/60			G 0 6 F 15/62	3 2 5 D

審査請求 未請求 請求項の数51 O L (全 61 頁)

(21) 出願番号	特願平7-341983	(71) 出願人	000005223 富士通株式会社 神奈川県川崎市中原区上小田中4丁目1番1号
(22) 出願日	平成7年(1995)12月28日	(72) 発明者	勝山 裕 神奈川県川崎市中原区上小田中1015番地 富士通株式会社内
(31) 優先権主張番号	特願平7-229508	(72) 発明者	直井 聡 神奈川県川崎市中原区上小田中1015番地 富士通株式会社内
(32) 優先日	平7(1995)9月6日	(74) 代理人	弁理士 大曾 義之 (外1名)
(33) 優先権主張国	日本 (J P)		

(54) 【発明の名称】 文書画像からのタイトル抽出装置および方法

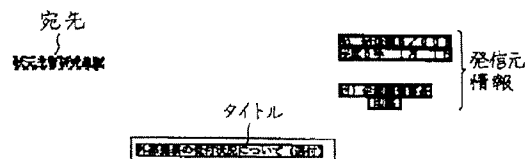
(57) 【要約】

【課題】 文書を画像データに変換して得られる文書画像から容易にタイトル部分を抽出することが課題である。

【解決手段】 タイトル抽出装置は、文書画像内の黒画素を走査し、それらが連結している領域に外接する矩形領域を文字矩形として抽出し、さらに、隣接する複数の文字矩形を統合して、それらの文字矩形に外接する矩形領域を文字列矩形として抽出する。次に、各文字列矩形の下線属性、枠付き属性、罫線属性等の属性と、文書画像内の文字列矩形の位置や相互の位置関係に基づいて、タイトルらしさのポイント計算を行い、高ポイントを獲得した文字列矩形をタイトル矩形として抽出する。また、表形式の文書の場合、表内からタイトル矩形を抽出することもできる。抽出されたタイトル矩形内の文字は、認識処理後に文書画像のキーワードとして用いられる。

タイトルおよび宛先・発信元情報の他の抽出結果を示す図

technical_news



【発明】 本発明は、図1に示すように、文書画像からタイトル部分を抽出する装置および方法に関する。図1は、本発明の装置の構成を示すブロック図である。図1に示すように、本発明の装置は、文書画像の入力を受け、文書画像を処理してタイトル部分を抽出する。抽出されたタイトル部分は、図2に示すように、文書画像の他の部分と区別して表示される。図2は、抽出されたタイトル部分の例を示す図である。図2に示すように、抽出されたタイトル部分は、文書画像の他の部分と区別して表示される。図2に示すように、抽出されたタイトル部分は、文書画像の他の部分と区別して表示される。

記

発明者： 〇 「外務省特許庁長官」 受付特許 (平成9年 8月20日) 1 第

上
[発明： 中村 〇〇〇〇〇〇]

【特許請求の範囲】

【請求項1】 文書を画像データに変換して得られる文書画像から必要とする部分領域を取り出して認識する情報処理装置において、

前記文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成する文字領域生成手段と、

前記文字領域生成手段が生成した1つ以上の文字領域を統合して、該1つ以上の文字領域を含む文字列領域を生成する文字列領域生成手段と、

前記文字列領域生成手段が生成した複数の文字列領域の属性に基づいて、該複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出するタイトル抽出手段とを備えることを特徴とするタイトル抽出装置。

【請求項2】 前記タイトル領域に含まれる文字領域を切り出して、文字認識を行う認識手段をさらに備えることを特徴とする請求項1記載のタイトル抽出装置。

【請求項3】 前記文字領域生成手段は、前記黒画素連結領域の外接矩形を前記文字領域として求め、前記文字列領域生成手段は、前記文書画像内における該外接矩形の辺の長さの頻度のヒストグラムを作成し、特定の頻度と該特定の頻度を与える最大の辺の長さとの関係表を作成し、該関係表を用いて、前記文書画像内において現れる頻度の高い第1の辺の長さを求め、該第1の辺の長さを基準として用いて前記文字列領域を生成することを特徴とする請求項1記載のタイトル抽出装置。

【請求項4】 前記文字列領域生成手段は、前記関係表において頻度が大きく変化する辺の長さをもとに、前記第1の辺の長さを決定することを特徴とする請求項3記載のタイトル抽出装置。

【請求項5】 前記文字列領域生成手段は、前記外接矩形の高さまたは幅を前記辺の長さとして用いて、前記ヒストグラムを作成することを特徴とする請求項3記載のタイトル抽出装置。

【請求項6】 前記文字列領域生成手段は、前記第1の辺の長さを用いて閾値を生成し、該閾値を用いて不要な文字領域を除去することを特徴とする請求項3記載のタイトル抽出装置。

【請求項7】 前記文字列領域生成手段は、前記閾値を用いて図表または写真の外接矩形を除去することを特徴とする請求項6記載のタイトル抽出装置。

【請求項8】 前記文字領域生成手段は、前記黒画素連結領域の外接矩形を前記文字領域として求め、前記文字列領域生成手段は、各外接矩形の第1の辺を用いて、前記文書領域内の第1の方向における外接矩形の分布範囲を表す第1のヒストグラムを作成し、該第1のヒストグラムの形状から外接矩形のグループを求め、該グループ毎に処理を行うことを特徴とする請求項1記載のタイトル抽出装置。

【請求項9】 前記文字列領域生成手段は、各外接矩形の第2の辺を用いて、前記文書領域内の第2の方向にお

ける外接矩形の分布範囲を表す第2のヒストグラムを作成し、前記第1および第2のヒストグラムの形状から前記グループを求め、該グループ毎に処理を行うことを特徴とする請求項8記載のタイトル抽出装置。

【請求項10】 前記文字列領域生成手段は、前記第1の辺の中線上に頂点を持つ二等辺三角形を作成し、該二等辺三角形を用いて前記第1のヒストグラムを作成することを特徴とする請求項8記載のタイトル抽出装置。

【請求項11】 前記文字列領域生成手段は、前記グループに属する外接矩形を探索して、重複する2つ以上の外接矩形を求め、該2つ以上の外接矩形を1つの外接矩形に統合することを特徴とする請求項8記載のタイトル抽出装置。

【請求項12】 前記文字列領域生成手段は、前記グループに属する外接矩形を探索して、ネストしている外接矩形を求め、ネストを除去することを特徴とする請求項8記載のタイトル抽出装置。

【請求項13】 前記文字列領域生成手段は、基準とする第1の外接矩形が属するグループ内の外接矩形を探索して、該第1の外接矩形に近接する第2の外接矩形を求め、該第1および第2の外接矩形の連結関係を表す連結関係表を作成し、該連結関係表を用いて前記文字列領域を生成することを特徴とする請求項8記載のタイトル抽出装置。

【請求項14】 前記文字領域生成手段は、前記黒画素連結領域の外接矩形を前記文字領域として求め、前記文字列領域生成手段は、基準とする第1の外接矩形に近接する第2の外接矩形を求め、該第1および第2の外接矩形の連結関係を表す連結関係表を作成し、該連結関係表を用いて該第1および第2の外接矩形に同じ識別情報を付加することにより、該第1および第2の外接矩形を1つの文字列領域に統合することを特徴とする請求項1記載のタイトル抽出装置。

【請求項15】 前記文字列領域生成手段は、前記第1の外接矩形から前記第2の外接矩形へ向かうポイントと、前記第2の外接矩形から前記第1の外接矩形へ向かうポイントのうち、少なくとも一方を前記連結関係表に格納することを特徴とする請求項14記載のタイトル抽出装置。

【請求項16】 前記文字列領域生成手段は、前記第1の外接矩形と前記第2の外接矩形の間に枠線がある場合には、該第1および第2の外接矩形を連結しないことを特徴とする請求項14記載のタイトル抽出装置。

【請求項17】 前記タイトル抽出手段は、前記文字列領域の内部を横方向に複数の部分領域に分割して、各部分領域の中で黒画素占有率の大きな部分線分領域を抽出し、閾値以上の高さの部分線分領域については高さを無視して、横方向に連結している各部分線分領域を統合し、統合された線分領域を抽出する線分抽出手段を有し、

該線分領域を用いて前記タイトル領域を抽出することを特徴とする請求項1記載のタイトル抽出装置。

【請求項18】 前記線分抽出手段は、前記文字列領域の内部を重複する複数の部分領域に分割することを特徴とする請求項17記載のタイトル抽出装置。

【請求項19】 前記線分抽出手段は、前記文字列領域の幅に近い長さの前記線分領域を抽出することを特徴とする請求項17記載のタイトル抽出装置。

【請求項20】 前記タイトル抽出手段は、前記線分領域が前記文字列領域内の下部にあるとき、該線分領域を下線と判別し、該文字列領域を前記タイトル領域の候補とすることを特徴とする請求項17記載のタイトル抽出装置。

【請求項21】 前記線分抽出手段は、前記文字列領域から同じ程度の左端座標および右端座標を持つ2つの線分領域を抽出し、該左端座標付近で縦方向の黒画素の第3のヒストグラムを作成し、該右端座標付近で縦方向の黒画素の第4のヒストグラムを作成し、第3および第4のヒストグラムのピークの高さが前記2つの線分領域の距離程度であれば、前記文字列領域内に枠線があると判別することを特徴とする請求項17記載のタイトル抽出装置。

【請求項22】 前記タイトル抽出手段は、前記複数の文字列領域の属性として下線属性または枠付き属性を抽出し、抽出した属性と各文字列領域の位置と文字列領域間の相対的位置関係とのうち少なくとも1つを用いて、各文字列領域にポイントを与え、高ポイントの文字列領域を前記特定の文字列領域とすることを特徴とする請求項1記載のタイトル抽出装置。

【請求項23】 前記タイトル抽出手段は、下線属性または枠付き属性を持つ文字列領域に一定の得点を与えることを特徴とする請求項22記載のタイトル抽出装置。

【請求項24】 前記タイトル抽出手段は、第1の方向の中心座標が前記文書画像の中央付近にある文字列領域に一定の得点を与えることを特徴とする請求項22記載のタイトル抽出装置。

【請求項25】 前記タイトル抽出手段は、上下にある文字列領域との距離が離れている文字列領域に一定の得点を与えることを特徴とする請求項22記載のタイトル抽出装置。

【請求項26】 前記タイトル抽出手段は、左側に他の文字列領域がないような文字列領域に一定の得点を与えることを特徴とする請求項22記載のタイトル抽出装置。

【請求項27】 前記タイトル抽出手段は、枠線を含む第1の文字列領域の内部に第2の文字列領域があり、該第1の文字列領域と第2の文字列領域が閾値以上に離れていないような一定の位置関係にある場合に、該第2の文字列領域が枠付き属性を持つとみなすことを特徴とする請求項22記載のタイトル抽出装置。

【請求項28】 前記タイトル領域の位置またはサイズの情報から、他の文字列領域の相対的な位置関係またはサイズを求め、該他の文字列領域の相対的な位置関係またはサイズが特定の条件を満たすとき、該他の文字列領域を宛先領域として抽出する宛先抽出手段をさらに備えることを特徴とする請求項1記載のタイトル抽出装置。

【請求項29】 前記タイトル領域の位置またはサイズの情報から、他の文字列領域の相対的な位置関係またはサイズを求め、該他の文字列領域の相対的な位置関係またはサイズが特定の条件を満たすとき、該他の文字列領域を発信元情報領域として抽出する発信元情報抽出手段をさらに備えることを特徴とする請求項1記載のタイトル抽出装置。

【請求項30】 前記タイトル抽出手段は、前記文書画像内で一定範囲のサイズの文字列領域が存在する文書領域を求め、該文書領域から前記タイトル領域を抽出することを特徴とする請求項1記載のタイトル抽出装置。

【請求項31】 前記タイトル抽出手段は、隣接した2つの文字列領域のサイズまたは座標値が類似している場合に、該2つの文字列領域を1つの文字列領域に統合することを特徴とする請求項1記載のタイトル抽出装置。

【請求項32】 前記タイトル抽出手段は、文字列領域のサイズまたは形状から罫線を表すと判定した時、該文字列領域に罫線属性を設定し、該罫線属性を用いて前記タイトル領域を抽出することを特徴とする請求項1記載のタイトル抽出装置。

【請求項33】 前記タイトル抽出手段は、前記罫線属性を持つ第3の文字列領域の上にある第4の文字列領域に下線属性を設定し、該第4の文字列領域を前記タイトル領域の候補とすることを特徴とする請求項32記載のタイトル抽出装置。

【請求項34】 文書を画像データに変換して得られる文書画像から必要とする部分領域を取り出して認識する情報処理装置において、

前記文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成する文字領域生成手段と、

前記文字領域生成手段が生成した1つ以上の文字領域を統合して、該1つ以上の文字領域を含む文字列領域を生成する文字列領域生成手段と、

一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、該表領域内の複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出するタイトル抽出手段とを備えることを特徴とするタイトル抽出装置。

【請求項35】 前記タイトル抽出手段は、第5の文字列領域の内部に罫線がある場合に、該罫線の位置で該第5の文字列領域を分割することを特徴とする請求項34記載のタイトル抽出装置。

【請求項36】 前記タイトル抽出手段は、前記第5の文字列領域内の複数の文字列領域の間に黒画素があるかどうかを調べ、黒画素がある位置で該第5の文字列領域を

分割することを特徴とする請求項35記載のタイトル抽出装置。

【請求項37】 前記タイトル抽出手段は、前記第5の文字列領域内の複数の文字領域と、前記文字領域生成手段が該第5の文字列領域内を対象にして再度求めた複数の文字領域との差異を調べ、該差異が検出された位置で該第5の文字列領域を分割することを特徴とする請求項35記載のタイトル抽出装置。

【請求項38】 前記タイトル抽出手段は、前記表領域内の第6の文字列領域の近くの特定領域に罫線があるかどうかを調べ、罫線がなければ該第6の文字列領域を表外の文字列領域とすることを特徴とする請求項34記載のタイトル抽出装置。

【請求項39】 前記タイトル抽出手段は、前記表領域内の文字列領域相互の位置関係から、上側に表内文字列領域がないような文字列領域を前記第6の文字列領域とし、該第6の文字列領域の上側にある前記特定領域の黒画素を探索し、一定閾値以上の黒画素が検出された場合に、該特定領域内に前記罫線があると判定することを特徴とする請求項38記載のタイトル抽出装置。

【請求項40】 前記タイトル抽出手段は、前記第6の文字列領域の上の他の文字列領域または表領域までの間を前記特定領域とし、前記閾値を該第6の文字列領域と該他の文字列領域または表領域との位置関係から決めることを特徴とする請求項39記載のタイトル抽出装置。

【請求項41】 前記タイトル抽出手段は、前記複数の文字列領域を前記表領域の左上に近い順に優先的に出力することを特徴とする請求項34記載のタイトル抽出装置。

【請求項42】 文字列領域生成手段は、前記1つ以上の文字列領域を含む文字列矩形を文字列領域として生成し、前記タイトル抽出手段は、該文字列矩形の特定の頂点の座標値をもとに、前記表領域内の複数の文字列矩形に優先順位を付けることを特徴とする請求項41記載のタイトル抽出装置。

【請求項43】 前記タイトル抽出手段は、前記複数の文字列領域のうち、項目らしい文字列領域を項目領域とし、タイトルらしい文字列領域を前記タイトル領域として、優先順位を付けて出力することを特徴とする請求項34記載のタイトル抽出装置。

【請求項44】 前記タイトル抽出手段は、あらかじめ決められた項目とタイトルの位置および文字数の関係に該当する文字列領域のペアを求め、該文字列領域のペアを上から順に出力することを特徴とする請求項43記載のタイトル抽出装置。

【請求項45】 前記タイトル抽出手段は、前記複数の文字列領域のうち閾値以上の文字数を持つ第7の文字列領域を、前記項目領域として出力することを特徴とする請求項43記載のタイトル抽出装置。

【請求項46】 前記タイトル抽出手段は、前記第7の

文字列領域の右側の文字列領域を、前記タイトル領域として出力することを特徴とする請求項45記載のタイトル抽出装置。

【請求項47】 前記タイトル抽出手段は、前記複数の文字列領域のうち、閾値未満の文字数を持つ第8の文字列領域を、前記項目領域として出力し、該第8の文字列領域の右側にあつて該閾値以上の文字数を持つ文字列領域を、前記タイトル領域として出力することを特徴とする請求項43記載のタイトル抽出装置。

【請求項48】 情報処理装置により用いられる記憶媒体であつて、該情報処理装置が、文書を画像データに変換して得られる文書画像内の連結した黒画素からなる黒画素連結領域を含む文字列領域を生成し、

1つ以上の文字列領域を統合して、該1つ以上の文字列領域を含む文字列領域を生成し、

複数の文字列領域の属性に基づいて、該複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出するように導くことを特徴とする記憶媒体。

【請求項49】 情報処理装置により用いられる記憶媒体であつて、該情報処理装置が、文書を画像データに変換して得られる文書画像内の連結した黒画素からなる黒画素連結領域を含む文字列領域を生成し、

1つ以上の文字列領域を統合して、該1つ以上の文字列領域を含む文字列領域を生成し、

一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、

該表領域内の複数の文字列領域のうち、特定の文字列領域をタイトル領域として抽出するように導くことを特徴とする記憶媒体。

【請求項50】 文書を画像データに変換して文書画像を生成し、

該文書画像内の連結した黒画素からなる黒画素連結領域を含む文字列領域を生成し、

1つ以上の文字列領域を統合して、該1つ以上の文字列領域を含む文字列領域を生成し、

複数の文字列領域の属性に基づいて、該複数の文字列領域のうち特定の文字列領域をタイトル領域として抽出し、

該タイトル領域に含まれる文字を認識することを特徴とするタイトル抽出方法。

【請求項51】 文書を画像データに変換して文書画像を生成し、

該文書画像内の連結した黒画素からなる黒画素連結領域を含む文字列領域を生成し、

1つ以上の文字列領域を統合して、該1つ以上の文字列領域を含む文字列領域を生成し、

一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、

該表領域内の複数の文字列領域のうち、特定の文字列領域をタイトル領域として抽出し、
該タイトル領域に含まれる文字を認識することを特徴とするタイトル抽出方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は画像データの認識処理に係り、文書を画像データとして取り込んだ文書画像から、タイトル部分の領域を抽出するタイトル抽出装置および方法に関する。

【0002】

【従来の技術とその問題点】一般文書をスキャナ等の光電変換装置で読み込んで得られる画像データである文書画像から、文書のタイトル等の部分領域を抽出する従来技術としては、以下に示す様なものがある。

(1) タイトルなどの領域が固定されている文書を対象として、固定領域をタイトルとして抽出する(特開昭64-46873)。

(2) 文書に色マーカや枠線で囲むなどの特定のマークを付けてから、光電変換装置で読み込んで、特定の色部分や特定のマーク部分の抽出によってタイトル部分を抽出する(特開平1-150974)。

(3) 文書の文字列や写真などの物理構造を本構造等に表現して、その論理構造とのマッチングをとることで、物理構造に「タイトル」、「著者名」等のタグ付けをする(特開平1-183784、特開平5-342326等)。

(4) 文書画像の一部の領域を指定し、その内部を投影して黒画素のヒストグラムを作成する。そのヒストグラム上で、投影した黒画素の値が2つの閾値の間にある部分が連続する範囲を求め、その連続長が閾値より大きな部分をタイトルとして抽出する(特開平5-274471)。

【0003】そのほかに、表を含む文書画像から表内のタイトル等の部分領域を抽出する従来技術として、以下に示す様なものがある。

(5) 表を含む固定フォーマットの文書(タイトルなどの領域が固定されている文書)を対象として、固定された領域をタイトルとして抽出する(特開平7-093348)。

(6) 文書画像を投影して黒画素のヒストグラムを作成し、ヒストグラムの分布から枠線を抽出して、枠線に囲まれる文字列をタイトルとして抽出する(特開平5-274367)。

(7) 文書画像内の全文字領域を文字認識し、得られた文字コードに対して単語(キーワード)照合や形態素解析等の言語的、論理的な知識処理を行い、その結果からタイトルらしい文字列を抽出する(特開平3-276260)。

(8) 文書画像内の白画素連結部分で囲まれた領域を表

部分として抽出し、その内部から罫線を抽出して、罫線で囲まれた領域を求める。そして、求めた領域の内部にある画像とあらかじめ決められた文字列(テンプレート)とのテンプレートマッチングを行うことで、それと同じ文字列をタイトルとして抽出する(特開平3-74728)。

【0004】しかしながら、これらの従来技術にはそれぞれ以下のような問題がある。

(1) および(5)の方法では、書式の固定した文書しか扱えない。書式を変更する場合は、抽出する部分の設定も変更する必要がある。

(2)の方法では、原稿文書にマークを付ける手間がかかる。

(3)の方法では、本構造等で表現した論理構造の辞書を用意する必要がある。また、辞書に無い論理構造の文書については、正確にタイトルを抽出することができなくなる。

(4)の方法では、文書画像の一部の領域の指定方法が明らかではないが、仮に全領域にこの方法を適用すると、図表などの大きな黒画素部分を誤ってタイトルとして抽出してしまう恐れがある。また文字だけの文書でも、文字サイズが大きな文字列がタイトルとは限らないので、誤抽出する可能性がある。

(6)単純な枠線に囲まれている表ならばこの方式でもよいが、実際には罫線が複雑に組み合わせられた表が使われることが多いため、そのような場合にタイトル領域を正確に特定できない。

(7)現在の文字認識処理ではかなりの処理時間がかかるため、実質的にバッチ処理としてしか使用方法がない。また、認識率は100%ではないので、タイトルの位置の情報を使わなければ、誤った部分をタイトルとして抽出することが多いと考えられる。

(8)画像上のテンプレートマッチングはマッチング処理自体に時間がかかるだけでなく、テンプレートのフォント形状またはサイズの影響を受けやすく、誤りやすいという欠点がある。また、この方法ではあらかじめ決まった文字列だけしかタイトルとして抽出できず、対象とする文書が限定される。

【0005】このように、従来のタイトル抽出方法では、ユーザにとって特別な準備作業や操作が必要であったり、対象とする文書やタイトルが限定される等の問題がある。

【0006】本発明は、文書画像から容易にタイトル部分を抽出することのできるタイトル抽出装置およびその方法を提供することを目的とする。

【0007】

【問題を解決するための手段】図1は、本発明のタイトル抽出装置の原理図である。図1のタイトル抽出装置は、文字領域生成手段1、文字列領域生成手段2、およびタイトル抽出手段3を備える。

【0008】文字領域生成手段1は、文書を画像データに変換して得られる文書画像内の連結した黒画素からなる黒画素連結領域を含む文字領域を生成する。文字列領域生成手段2は、文字領域生成手段1が生成した1つ以上の文字領域を統合して、それらの文字領域を含む文字列領域を生成する。

【0009】タイトル抽出手段3は、文字列領域生成手段2が生成した複数の文字列領域の属性に基づいて、それらの複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出する。

【0010】文字領域生成手段1は文書画像内の黒画素を走査し、例えば、それらが連結している領域に外接する矩形領域を文字領域として抽出する。この結果、文書内の多数の文字に対応する多数の文字領域が生成される。

【0011】次に、文字列領域生成手段2は、隣接する複数の文字領域を統合して、例えば、それらの文字領域に外接する矩形領域を文字列領域として抽出する。この文字列領域は、例えば、横書き文書内の1行分の文字列に対応する。

【0012】タイトル抽出手段3は、生成された各文字列領域の下線属性、枠付き属性、罫線属性等の属性に基づいてタイトルらしさを評価し、最もタイトルらしいと考えられる特定の文字列領域を、タイトル領域として抽出する。

【0013】ここで、下線属性とは、文字列領域の内部または下方に下線があることを意味し、下線フラグ等を用いて表現される。枠付き属性とは、文字列領域のまわりを枠線が囲んでいることを意味し、枠線フラグ等を用いて表現される。また、罫線属性とは、文字列領域が横長または縦長の罫線に対応することを意味し、罫線フラグ等を用いて表現される。下線属性や枠付き属性を持つ文字列領域は文書のタイトルである可能性が高く、罫線属性を持つ文字列領域はタイトルである可能性がほとんどない。そこで、このような属性をもとにタイトルらしさを自動的に評価することができる。

【0014】また、タイトル抽出手段3は、一定以上の大きさを持つ黒画素連結領域を含む表領域を抽出し、該表領域内の複数の文字列領域のうち特定の文字列領域を、タイトル領域として抽出する。

【0015】表領域としては、例えば、黒画素連結領域に外接する矩形領域のうち、一定の閾値以上の大きさを持つものが用いられる。そして、タイトル抽出手段3は、表領域内の文字列領域相互の位置や文字数等の関係を調べて、タイトルらしさを評価し、最もタイトルらしいと考えられる特定の文字列領域を、タイトル領域として抽出する。

【0016】例えば、表領域の左上に近い文字列領域ほどタイトルらしいと考えられ、また、文字数の大きな文字列領域もタイトルらしいと考えることができる。本発

明のタイトル抽出装置によれば、表形式文書を含む様々な文書画像を対象として、原稿にマークすることもなく、特別な構造辞書が不要で、文字サイズのみに影響されない、高精度なタイトル抽出処理を行うことができる。また、抽出されたタイトル領域に含まれる文字領域を切り出して文字認識を行い、認識結果を文書画像のキーワードとして用いることもできる。

【0017】図1の文字領域生成手段1、文字列領域生成手段2、およびタイトル抽出手段3は、例えば、実施の形態における図2のプロセッサ14に相当する。

【0018】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態を詳細に説明する。最近、従来より紙媒体で保存してきた情報を電子化する動きが多く見られる。その中の1つに電子ファイリングシステムがある。電子ファイリングシステムにおいては、紙文書がイメージスキャナ等の光電変換装置で画像に変換され、それに検索のためのキーワードや管理情報が付与されて、光ディスクやハードディスクに保存される。

【0019】このような方法では文書を画像データとして保存するため、文書に記されているすべての文字を文字認識技術でコード化してから保存する方法よりも、ディスク容量は多く必要となる。その反面、手軽で処理速度が速く、文字以外の絵や表などもそのまま保存できるメリットがある。しかし、保存された情報を検索するために、文書画像と共にキーワードや番号などの管理情報を付与しなければならない。従来のシステムはこのキーワード付けの手間がかかり、使いづらいシステムであった。

【0020】この使いづらさを解決するために、文書中にあるタイトル部分をキーワードとみなしてそれを自動的に抽出し、その部分を文字認識してコード化した結果を文書画像と共に保存する方法が考えられる。

【0021】現在の文字認識の処理速度は速くても数十文字/秒であり、A4の文書1枚を処理するには30秒から数分の処理時間がかかってしまう。したがって、タイトル抽出を高速化するためには、文書全体を文字認識するのではなく、まず画像上で必要なタイトル部分だけを抽出した後にそれを文字認識する方法が有力である。

【0022】また、文書中の全文字を認識して論理的にタイトルを抽出する方法では、画像上におけるタイトル部分の位置関係が考慮されない。このため、誤認識や文章のつながりの影響で、正確にタイトルコードを抽出できない場合がかなりあるはずである。

【0023】このように、電子ファイリングシステムの効率的な運用を考えると、文書画像から直接タイトル部分(領域)を抽出する技術は、重要な技術である。そこで、電子ファイリングシステムを例にとり、本発明のタイトル抽出技術について説明する。

【0024】図2は、実施形態のタイトル抽出システムの構成図である。図2のタイトル抽出システムは、電子ファイリング装置11、光電変換装置12、ディスプレイ端末13、プロセッサ14、およびメモリ15を備え、これらの装置はバス16により結合されている。

【0025】電子ファイリング装置11は、ハードディスクや光ディスク等の格納装置を備え、複数の画像データを個別に格納している。光電変換装置12は、例えばスキャナ等の光学的な読み取り装置であり、文書や絵、写真等を画像データに変換する。こうして取り込まれた画像データは、電子ファイリング装置11またはメモリ15に格納される。ディスプレイ端末13は、ディスプレイ装置とキーボードやマウス等の入力装置とを備えたオペレータ端末である。

【0026】プロセッサ14は、ディスプレイ端末13から入力された指示に従い、光電変換装置12からメモリ15に取り込んだ文書画像、または、電子ファイリング装置11から取り出した文書画像からタイトル等の特定領域を抽出する。そして、抽出した領域に含まれる文字を認識する。尚、文字の認識処理は、タイトル抽出システムの外部のシステムにより行う構成としてもよい。

【0027】図2のタイトル抽出システムは、例えば、後述する図4に示すような文書画像から図6に示すような文字の外接矩形を求め、さらに複数の文字の外接矩形を統合して、図27に示すような文字列矩形を求める。そして、各文字列矩形が文書の中で強調されているかどうかを調べる。

【0028】例えば、図14に示すような枠線で囲まれている文字列は強調されているものとみなし、それだけでタイトルらしいと考えられるので、それらをタイトル候補として抽出する。そのほかにも、下線を持つ文字列や大きな文字列は強調文字列と考えて、タイトル候補として抽出する。また、文書内での文字列の位置や隣接する他の文字列との位置関係も、タイトル文字列を識別するために有力な情報として用いられる。

【0029】このように、文字列が強調されているかどうかやその位置等の外見的情報をもとにタイトル候補の文字列を選択するので、タイトルである蓋然性の高い領域を文書画像から簡単に抽出することができる。この抽出方法は文書全体を認識してからタイトルを抽出する方法よりも高速であり、対象とする文書を選ばないという点で汎用的である。また、文字列の2つ以上の外見的情報を組合せて用いることにより、比較的正確にタイトル領域を特定することができる。

【0030】図3は、図2のタイトル抽出システムによるタイトル抽出処理のフローチャートを示している。図3の処理においては、前提条件として横書き文書を対象としているが、横書きに限らず縦書き文書でも横書きと同様の処理で対応可能である。縦書き文書の場合には、文字領域や文字列領域の高さと幅が、横書き文書の場合

と互いに逆の役割を果たすことになる。

【0031】図3において処理が開始されると、光電変換装置12が文書を読み取り、画像データ（文書画像）としてメモリ15に格納する（ステップS1）。このとき、処理の高速化のために、読み取った原画像を縦横それぞれ1/8に圧縮して圧縮画像を作成し、それを文書画像としてメモリ15に格納しておく。

【0032】画像を圧縮する際には、線分がとぎれないように論理ORの圧縮方法を用いる。つまり、原画像の8×8画素の領域の中に1つでも黒画素があれば、圧縮画像の対応する画素を黒とし、全く黒画素がなければそれを白とする。

【0033】次に、プロセッサ14が、文書画像から文字列（あるいは行）を抽出し、文字列の外接矩形（文字列矩形）を求め、その座標をメモリ15に保存する（ステップS2）。次に、保存した文字列矩形から、横幅が小さい矩形や縦長矩形をノイズ矩形として除去し（ステップS3）、さらに文字列らしくない矩形を除いて、文書領域を決定する（ステップS4）。

【0034】次に、残った文字列矩形を縦方向（y座標）で並べ替え（ステップS5）、枠の画像を含む矩形（枠矩形）を抽出して、枠矩形内にある文字列矩形を枠付き矩形としてマークする（ステップS6）。また、下線の画像を含む矩形を抽出して、そのすぐ上にある文字列矩形を下線矩形としてマークする（ステップS7）。

【0035】次に、タイトルらしさのポイント計算を行って、ポイントの高い文字列矩形をタイトルとして抽出し（ステップS8）、その結果を用いて文書の宛先と発信元情報を抽出する（ステップS9、S10）。そして、抽出したタイトル、宛先、発信元情報の認識処理を行って（ステップS11）、処理を終了する。

【0036】次に、一般的な社内文書を例に取り、タイトル抽出処理を詳細に説明する。社内文書には、通常、「タイトル」、「宛先」、「発信日」、「発信所属」、「発信管理番号」、「本文（図表あり）」などの要素が含まれており、それらが様々な配置で記載されている。ここでは、このような様々な書式の文書からタイトル、宛先、および発信者情報（発信日、発信所属、発信管理番号等）を抽出する。

【0037】図4は、スキャナでメモリ15に読み込まれた文書画像の例を示している。図4の文書画像は、ソフトウェア販推レポートの送付表に関するものであり、この文書のタイトルは「ソフトウェア販推レポート 送付表」で、その下に宛先や発信元情報が記載されている。プロセッサ14は、まずこの文書画像から文字列を抽出する。図5は、図3のステップS2の文字列抽出処理のフローチャートである。

【0038】図5において処理が開始されると、プロセッサ14は、まず文書画像から文字に相当する矩形を抽出する。そのために、文書画像に対してラベリングによ

る黒画素連結処理を施し、黒画素の外接矩形を求めて保存する(ステップS21)。

【0039】ここでは、2値化されている圧縮画像の黒画素を8連結で走査し、連結がある場合にはそれらの黒画素に同一のラベル値を与えることによって黒画素連結領域を生成し、その外接矩形(文字矩形)を求める。8連結による走査とは、1つの黒画素の上、下、左、右、左上、右上、左下、右下の8方向を走査して、他の隣接黒画素があるかどうかを調べる処理を意味する。求められた外接矩形は、ファイルlbtblに保存される。図4の文書画像にラベリング処理を施した結果は、図6のようになる。

【0040】次に、ラベリングにより得られた外接矩形の高さの頻度分布を表すヒストグラムを求め、高さの最頻値freqを求める(ステップS22)。ここでは、まずラベリング結果の外接矩形の集合lbtblから、図7に示すような矩形高さのヒストグラムを作成する。図7において、横軸が各外接矩形の高さを表し、縦軸がその高さを持つ矩形の数(頻度値)を表す。外接矩形の高さは、例えば1画素の高さを単位高さとして求める。

【0041】次に、頻度値と、その頻度値を持つ矩形高さの内で最大の高さとの対応関係を求め、矩形高さテーブルheightに保存する。そして、heightの中を頻度値0から順に調査していき、高さの変化が1以内で頻度値の変化するものが連続し、それらの頻度値の変化の合計が9以上の場合に、それらの連続する高さのうちで最も高いものを矩形高さの最頻値freqとする。

【0042】図8は、図7のヒストグラムに対応するheightの内容を表すヒストグラムを示している。図8において、頻度値が急激に変化する高さがfreqとなっていることがわかる。このようにしてfreqを求めておけば、1文字よりも小さなノイズの影響を排除することができる。

【0043】図9は、heightの簡単な例を示している。図9においては、4つの頻度値と、各頻度値を持つ矩形高さの内で最大の高さとが、それぞれペアで格納されている。freqを求めるために、このheightの内容をヒストグラムにすると図10のようになる。図10のヒストグラムを、頻度値の低いところから、つまり高さの高いところから順に見ていくと、高さが10、9、8の位置で、頻度値がそれぞれ5、5、7だけ変化していることが分かる。これらの連続する高さの差は1であり、頻度値の変化の合計は17である。したがって、高さ10、9、8において頻度値の変化の合計が9以上となっているので、それらの中で最初に現れた高さ10をfreqとする。

【0044】次に、枠線や図表の外接矩形を除去するために、大きな矩形と判断するための閾値を設定し、それより大きな矩形を抽出する。そして、抽出した大きな矩形の中から枠線を含む矩形を抽出して保存する(ステッ

プS23)。

【0045】ここでは、例えば、freqより大きな矩形で最も頻度値が大きな矩形高さを大きな矩形の閾値th_largeとし、th_largeより大きな矩形を抽出して、ファイルboxに保存する。

【0046】次に、boxの中の大きな矩形から枠線を抽出するために、それぞれの大きな矩形の内部を図11に示すように縦に部分分割して、重複のある短冊状の部分領域を作る。そして、各短冊状部分領域の中で、一定割合以上の黒画素占有率を持つ高さ1画素の横方向の線状領域を求める。さらに2つ以上の線状領域が上下に連続していれば、それらを統合して1つにまとめた部分線分を求める。

【0047】図12は、図11の大きな矩形の中の1つの短冊状部分領域を示している。図12において、幅wの部分領域は高さ1の線状領域に細分され、一定割合以上の黒画素を含み、上下に連続する線状領域が1つの部分線分矩形として統合されている。図12に示すように、1つの部分領域には2つ以上の部分線分矩形が存在する可能性がある。次に、このようにして求めた部分線分矩形同士が左右で8連結の関係にあれば、それらを1つの線分として扱う。図13(a)、(b)、(c)は、それぞれ8連結の関係にある2つの部分線分矩形の例を示している。こうして、図11の場合は、大きな矩形の上端部分から横方向の線分矩形が抽出される。

【0048】このようにして求めた線分矩形が、対象となる大きな矩形の横幅に比べて一定比率以上の場合に、長い線分矩形として抽出する。この長い線分矩形の両端と大きな矩形の両端の差が一定マージン以内にあり、かつ、長い線分矩形の上下端のy座標と大きな矩形の上下端のy座標の差が矩形幅の一定比率よりも小さい時に、大きな矩形の上下に位置している横罫線と判断する。

【0049】そして、この大きな矩形の左右端付近の黒画素を縦方向に投影した頻度分布(ヒストグラム)を求め、そのピークの高さが矩形高さの一定比率より大きい場合に、左右端に縦罫線もあると判断する。このとき、この大きな矩形は枠線の外接矩形(枠矩形)と識別される。boxの中の各大きな矩形について同様の処理を行い、枠矩形のみをboxに残す。図14は、検出された枠矩形を示している。

【0050】次に、ラベリングで求めた外接矩形の集合lbtblから枠矩形および図表と判定された矩形(図表矩形)を除去して、結果を保存する(ステップS24)。ここでは、まずlbtblの中からboxに保存されている枠矩形を除去する。さらに、次のいずれかに該当する矩形を図表矩形と推定して、これらをlbtblから除去する。

- (a) 文書画像全体の高さの1/3より大きな矩形
- (b) 高さがfreqの3倍より大きく、高さ/幅の比が0.4より小さい矩形
- (c) 高さがfreqの3倍より大きく、文書画像全体の幅

の $1/3$ より大きな矩形

そして、除去後の矩形集合をnewtblとして管理する。この矩形集合newtblから文字列の外接矩形が抽出される。

【0051】newtbl内の矩形の中には、矩形同士が重複またはネストしているものも含まれている。このような矩形は1つにまとめた方が、矩形相互の位置関係が明確になり、文字列の抽出を効率的に行うことができる。そこで、newtblを対象にして、矩形同士が重複またはネストしているものを統一して、重複/ネストを除去し、結果をファイルlbtbl2に保存する(ステップS25)。

【0052】図15は、重複する2つの矩形の例を示している。図15において、矩形21と矩形22は、それぞれ右上がりの斜めの線分の外接矩形を表し、斜線部分で互いに重複している。この場合、矩形21、矩形22を、これらの矩形を包括する1つの矩形23にまとめて、重複を除去する。図16は、ネストしている複数の矩形を示している。図16において、矩形25、26、27は、矩形24に完全に内包されており、その矩形にネストしている。この場合、矩形24のみを残して、ネストを除去する。

【0053】ところで、newtblの中で、1つの矩形と重複/ネストしている他の矩形を探索するのには、次の2つの方法がある。

(d) 1つの矩形を基準として、残りの矩形全部を探索範囲とする。

(e) 縦または横方向に、矩形の辺の中線上に頂点を持つ二等辺三角形を作り、そのヒストグラムを作成する。そして、ヒストグラム中のそれぞれの頻度の山を構成する矩形の集合(グループ)を同時に記録する。ヒストグラム中で、山と山の距離が閾値より近いものを統合し、同時に対応する矩形集合も統合する。この矩形の集合を1つの探索範囲とし、この集合内にある矩形を基準とした場合は、この集合内を探索する。また、縦方向および横方向で作成した各矩形集合の重なり部分を求めて、探索範囲としてもよい。

【0054】図17は、(e)の方法で用いる二等辺三角形のヒストグラムの例を示している。図17において、矩形31、32の二等辺三角形36、37が1つの山41に投影され、矩形33の二等辺三角形38が山42に投影され、矩形34、35の二等辺三角形39、40が山43に投影されている。例えば、これらの山41、42、43が一定距離内にある場合は、矩形31、32、33、34、35は1つの矩形集合に統合される。あるいはまた、矩形31と矩形32のように、対応する二等辺三角形が1つの山に投影されるような矩形を1つの矩形集合にまとめてもよい。

【0055】(e)の方法によれば、限られた範囲の矩形のみを探索すればよいので、一般に(d)の方法よりも高速処理が可能である。図18は、このようにして重複/ネストが除去された外接矩形を示している。

【0056】次に、重複/ネスト除去後のlbtbl2に含まれる矩形の高さのヒストグラムを求め、高さの最頻値freq2を求める(ステップS26)。高さのヒストグラムの作成方法およびfreq2を求める方法は、ステップS22と同様である。

【0057】次に、lbtbl2から罫線矩形を抽出して、マークする(ステップS27)。ここでは、lbtbl2内で、高さがfreqの $1/2$ より小さく、幅がfreqの3倍より大きく、高さ/幅の比が 0.1 より小さい矩形を、罫線矩形としてマークする。

【0058】次に、1つの文字列に属する複数の文字を見つけるために、lbtbl2内の矩形相互の関係を求め、連結関係表connectに保存する(ステップS28)。ここでは、lbtbl2内の各矩形から上下左右に最も近い矩形を探索し、その結果をconnectに格納する。矩形相互の関係とは、ある基準矩形から上下左右の矩形への各ポイントと、上下左右の矩形から基準矩形へ向かう各ポイント、および基準矩形から上下左右の矩形までの距離を意味する。

【0059】図19は、1つの矩形を基準矩形とした場合の矩形間の連結関係を示している。図19において、上矩形は基準矩形の上に近接する矩形を表し、ポイント51、52により基準矩形と連結されている。上矩形は基準矩形の上に近接する矩形を表し、ポイント51、52により基準矩形と連結されている。下矩形は基準矩形の下に近接する矩形を表し、ポイント53、54により基準矩形と連結されている。左矩形は基準矩形の左に近接する矩形を表し、ポイント55、56により基準矩形と連結されている。右矩形は基準矩形の右に近接する矩形を表し、ポイント57、58により基準矩形と連結されている。

【0060】このようなポイントを格納する連結関係表connectの構造は、例えば図20に示すようになる。図20の連結関係表には、基準矩形のラベル値に続いて、上矩形へのポイント、上矩形からのポイント、下矩形へのポイント、下矩形からのポイント、左矩形へのポイント、左矩形からのポイント、右矩形へのポイント、右矩形からのポイントが格納されている。connectには、これらのポイントの他に基準矩形から上下左右の各矩形までの距離も格納される。

【0061】connectを作成する時には、枠矩形の4辺で連結関係が切断されるように設定しておく。これは、後に枠線を越えて文字列を抽出しないようにするためである。基準矩形に最も近い矩形を探索する場合も、ステップS25で用いた(d)と(e)の2通りの方法がある。

【0062】次に、スキャナによる読み取り時のノイズに相当するノイズ矩形を識別し、それと他の矩形との横の関係を切断する(ステップS29)。ここでは、矩形高さ、幅がfreqの $1/4$ より小さい矩形、または、高

／幅の比が0.1より小さいか1.0より大きくかつ上下の矩形との距離が一定値より大きい矩形を、ノイズ矩形と判定する。そして、その矩形と他の矩形との間の横方向のポイントを削除して、連結関係を切断する。

【0063】次に、隣接する矩形間の距離が離れている場合、または隣接する矩形間で大きさに差がある場合、それらの矩形の連結関係を切断する（ステップS30）。ここでは、基準矩形が次のいずれかの条件に該当する場合に、隣の矩形との連結関係を切断する。

(f) 基準矩形と隣の矩形との距離が、freq2 の3倍よりも大きい。

(g) 基準矩形または隣の矩形が、freq2 の3倍以上の大きさを持つ。

(h) 隣の矩形がfreq2 の2倍より大きい。

【0064】次に、文字矩形の集合lbtbl2とその連結関係connect から、文字列を抽出し、文字列の外接矩形（文字列矩形）を保存する（ステップS31）。ここでは、まず、lbtbl2内の矩形のうち、その矩形へ左から向かうポイントが無いもの、つまり、左側に矩形が無いものを開始矩形とする。次に、その矩形の識別番号（例えばラベル値）を右側にある他の矩形へ順次伝搬させていき、同じ識別番号を付加した複数の矩形を統合して、それらの外接矩形を文字列矩形とする。この時、開始矩形の識別番号を、抽出した文字列の識別番号（ラベル値）としてline_lab に保存しておく。そして、右側に連結する矩形が無くなったら伝搬を終了する。

【0065】図21は、こうして抽出された文字列矩形の例を示している。図21では、横に並んだ4つの文字矩形がラベル値L1を付加されて、1つの文字列矩形に統合されている。この場合、その文字列矩形のラベル値もL1になる。

【0066】もし、右側の矩形の識別番号が、既にline_lab 内にある文字列識別番号に一致する場合は、これまで伝搬してきた矩形の集合の識別番号を、右側の文字列識別番号へ置き換える。そして、置き換え前の文字列の識別番号はline_lab から除去しておく。

【0067】この処理の後、左から向かうポイントが無い矩形を再び検出し、これを基準矩形とする。その基準矩形の左側に矩形がある場合は、その左側の矩形は既に抽出された文字列の識別番号に組み込まれているはずである。そこで、その番号を基準矩形より右にある矩形に対して、右に連結した矩形が無くなるまで伝搬させ、識別番号を置き換える。そして、line_lab から置き換え前の矩形の番号を除去する。

【0068】例えば、図22に示すように、1つの文字列矩形の中に他の文字列矩形が存在する場合を考える。左からのポイントを持たない矩形64を基準矩形として、その左へのポイントを進ると左側に矩形61があることが分かる。矩形61は既にラベル値L0を持っているので、この値を矩形64、65へ伝搬させて、それら

のラベル値をL0に置き換える。こうして、ラベル値L5はline_lab から除去され、矩形61、62、63、64、65は1つの文字列矩形に統合される。

【0069】ここまでの処理において、同一文字列と識別された各矩形には同じ文字列識別番号がついている。そこで、全部の矩形を走査して、同じ文字列識別番号が付いている複数の矩形の座標から最左端、最右端、最上端、最下端を求め、それらを文字列矩形の外周を構成する座標として、ファイルlineに保存する。また、抽出した文字列の数をmaxline として保存する。

【0070】以上で、文字列抽出処理が終了する。図23は、こうして抽出された文字列矩形を示している。次に、プロセッサ14は、抽出された文字列矩形に対して、図3のステップS3からS7までの処理に対応する文字列矩形加工処理を施す。文字列矩形加工処理においては、各文字列矩形の下線属性、枠付き属性、罫線属性等の属性を抽出し、それらを記録する。後のポイント計算において、下線属性や枠付き属性を持つ文字列矩形にはより高いポイントが付与され、罫線属性を持つ文字列矩形にはより低いポイントが付与される。

【0071】図24は、この文字列矩形加工処理のフローチャートである。図24において処理が開始されると、まず横幅が小さい文字列矩形や縦長の文字列矩形をノイズ文字列矩形として除去し、結果を保存する（ステップS41）。ここでは、横幅がfreq/4より小さい文字列矩形、または、高さがfreq/4より小さくかつ高さ／幅の比が0.1より大きい文字列矩形をノイズとみなして除去し、残ったものをファイルline2 に保存する。図25は、ノイズ除去後の文字列矩形を示している。

【0072】次に、line2 内の文字列矩形間の接続関係を表す連結関係表 str_connを作る（ステップS42）。ここでの接続関係は、図19に示した文字矩形間の連結関係と同様のものを表し、str_connは図20に示した連結関係表と同様の構造を持つ。

【0073】次に、位置関係や高さが一定の条件を満たす2つ以上の文字列を統合してより長い文字列を求め、結果を保存する（ステップS43）。ここでは、次のいずれかに該当する場合に、それらの文字列矩形を1つに統合し、さらに大きな文字列矩形を求める。

(i) 文字列矩形間の距離が文字列矩形の高さより小さい場合

(j) 横方向に重複していて、高さがほぼ等しい文字列矩形

(k) 矩形高さの最頻値freq位の高さで、他の文字列矩形に完全に含まれる文字列矩形

(l) 3連の文字列矩形で両端の矩形のy座標がほぼ等しく、それらの間にある矩形だけが異なる場合

図26(a)、(b)、(c)、(d)は、それぞれ(i)、(j)、(k)、(l)の場合に統合されてできる文字列矩形の例を示している。このような処理を文

字列矩形の数が変化しなくなるまで繰り返し、残った文字列矩形をファイルline3に保存する。図27は、こうして文字列矩形を統合した結果を示している。図25と図27を比べると、例えば、文字列矩形「ソフトウェア販推レポート」と「送付表」とが、文字列矩形「ソフトウェア販推レポート 送付表」に統合されていることが分かる。

【0074】次に、文字列の高さのヒストグラムを作成し、文字列高さの最頻値 str_freq を求める（ステップS44）。ここでは、文字列矩形の高さのヒストグラムを、図7と同様にして作成する。そのヒストグラムから、 $freq2$ 以上で最大頻度を与える高さを求め、それを文字列矩形の高さの最頻値 str_freq とする。もし、最大頻度を与える高さが複数個あったら、 $freq2$ に近い方の高さを採用する。文字列矩形の高さのヒストグラムにおいて、 str_freq から連続する頻度分布を見ていったとき、 str_freq の両側に頻度値が0になる位置がある。これらの頻度値が0になる位置の直前の高さのうち、小さい方を st_h 、大きい方を en_h とする。

【0075】次に、ノイズを除いた文書領域を求めて、その領域の座標を保存する（ステップS45）。ここでは、文書画像の左右端にある一定領域内に一部分でも掛かるような文字列矩形は対象外として、高さが st_h 以上、 en_h 以下で、かつ、横幅が str_freq 以上で、かつ、高さ／幅の比が0.5未満の文字列矩形が存在する範囲を文書領域とする。そして、その領域の左端のx座標、上端のy座標、右端のx座標、下端のy座標を、それぞれ st_x 、 st_y 、 en_x 、 en_y として保存する。左右端の一定領域を無視するのは、例えばA4サイズの画像領域にB5版の本の1ページ分の画像を読み込んだような場合に、文書画像の左右に存在する隣のページの文字列矩形を、ノイズとして除去するためである。図28は、こうして求められた文書領域を示している。

【0076】次に、line3内の文字列矩形を縦方向（y座標）で並べ替える（ステップS46）。次に、line3内の文字列矩形間の連結関係を表す連結関係表 str_conn2 を作る（ステップS47）。この時、枠矩形を跨いで連結する関係がないようにする。

【0077】次に、各文字列矩形が枠矩形に完全に含まれているかどうかをチェックし、含まれている場合にはその文字列矩形に枠付きフラグを立てる（ステップS48）。ここでは、line3内の各文字列矩形に対して、それがboxに保存された枠矩形に完全に内包される場合に枠付き矩形とみなし、その文字列矩形に枠付きフラグを立てる。枠付き矩形の判定基準としては、枠矩形の内部にある文字列矩形をすべて枠付き矩形とみなす場合と、枠矩形と内部の文字列矩形の座標値が閾値以上に離れていない場合のみ枠付き矩形とみなす場合とがある。

【0078】次に、line3内の文字列矩形の中で、罫線矩形と判断したものに罫線フラグを立てる（ステップS

49）。ここでは、 str_freq の $1/2$ 以下で、高さ／幅の比が0.8より小さいか、または12.5より大きいものを罫線矩形とみなして、その文字列矩形に罫線フラグを立てる。

【0079】次に、line3内の文字列矩形を調べてその直下に下線らしい罫線矩形（下線矩形）が有る場合、または、文字列矩形内部を走査して内部に下線が有る場合は、その文字列矩形に下線フラグを立てる（ステップS50）。ここでは、罫線矩形の上に文字列矩形があり、それらの間の距離が str_freq より小さい範囲にあり、かつ、上の文字列矩形と罫線矩形の左右端の差が str_freq 以下のとき、上にある文字列矩形に下線フラグを立てる。図29は、下線矩形の例を示している。図29において、文字列矩形71の下には罫線フラグが立てられた横長の罫線矩形72があるため、これが下線矩形とみなされ、文字列矩形71には下線フラグが立てられる。

【0080】また、幅または高さが str_freq の $1/2$ 以上の文字列矩形を対象として、後に述べる方法で線分を抽出する。そして、文字列矩形内で抽出した線分が、文字列矩形の左右端から一定画素数の範囲にあり、かつ、線分の高さが矩形高さの $WAKUTHIN$ 倍（例えば0.3倍）以下で、かつ、線分の下y座標が矩形の下y座標から $str_freq/2$ だけ上の位置より下にあり、かつ、線分の上のy座標と矩形の上のy座標の差が $str_freq - 2$ よりも大きく、かつ、線分の下y座標と矩形の下y座標の差が線分の上のy座標と矩形の上のy座標の差よりも小さい場合に、この線分を文字列矩形内の下線として識別し、その文字列矩形に下線フラグを立てる。

【0081】こうして、文字列加工処理を終了する。図30は、枠付きフラグ、罫線フラグ、下線フラグを立てる処理を終えた後の文字列矩形を示している。図30において、L0～L54は、各文字列矩形に付加されたラベル値を表している。これらの文字列矩形のうち、ラベル値L1、L2、L16を持つ文字列矩形が枠付き矩形に相当する。

【0082】次に、図24のステップS50で文字列矩形から線分を抽出する方法を詳細に説明する。図31は、線分抽出処理のフローチャートである。図31において処理が開始されると、プロセッサ14は、まず文字列矩形を一定画素幅wの短冊状の部分領域に分割する（ステップS61）。この部分領域は、図11の場合と同様に半分ずつ重なるような領域とする。

【0083】次に、各部分領域の内部を上から下へ順に、縦1画素×横w画素の線状領域毎に注目していく。ある線状領域の内部の黒画素数が閾値よりも大きい場合に、この線状領域の内部が全て黒画素であるとみなし、これを黒領域とする。黒領域の直下に別の黒領域がある場合は、2つの黒領域は連続しているものと判断し、1つの黒領域（部分線分矩形）として扱う（ステップS6

2)。すなわち、黒領域を表す座標は、左右は部分領域の左右の座標、上は、上から順に走査していったときに白領域から黒領域へ変化するときの黒領域のy座標、下は、黒領域から白領域へ変化するときの黒領域のy座標となる。この結果、1つの部分領域から1つあるいは複数個の黒領域の座標が求められる。この操作を全部分領域で行い、黒領域の集合を求める。

【0084】次に、黒領域の中で高さが閾値より大きいものをワイルドカードと呼ぶことにする(ステップS63)。ワイルドカードは、例えば、文字列矩形内で文字が潰れて黒画素の塊になっているような場合に発生する。図32は、部分領域に分割された文字列矩形と、その中のワイルドカードの例を示している。また、図33は、1つの部分領域の中の線状領域とワイルドカードの例を示している。図33において、部分領域は15個の線状領域からなり、それらのうち上から12個の線状領域がワイルドカードを形成している。

【0085】次に、黒領域の集合を走査し、重複または隣接するものを統合して横長の矩形領域を求める(ステップS64～S69)。まず、最初に黒領域の集合から1つの黒領域を選び、それに注目する(ステップS64)。その黒領域がワイルドカード矩形でない場合は、その黒領域の上下端の座標と左右端の座標を、横長の矩形領域の座標として保存する。1回でも黒領域の集合から取り出した黒領域は、使用済みフラグを立てて二度と使用しない。

【0086】次に、黒領域の集合から1つの黒領域を取り出し、既に使用済みのものでなければ、記憶した横長矩形の座標と比較して、その右側に隣接または重複する関係にあるかどうかをチェックし、そのような関係にある黒領域を選ぶ(ステップS65)。そして、その黒領域がワイルドカードかどうかを判定し(ステップS66)、ワイルドカードの場合はその高さを無視して横方向に領域を統合する(ステップS67)。このとき、記憶している横長矩形の右端の座標を、そのワイルドカード矩形の右端の座標で置き換える。

【0087】右側に隣接または重複する黒領域がワイルドカードでない場合は、両方の矩形の上下座標を比較し、それらの差が閾値以内であれば、縦方向と横方向に領域を統合する(ステップS68)。このとき、右側のワイルドカードでない黒領域の上下座標を新しい横長の矩形領域の上下座標とする。また、黒領域の右端の座標を横長矩形の右端の座標とする。そして、黒領域をすべて調べたかどうかを判定し(ステップS69)、未処理の黒領域があれば、ステップS65以降の処理を繰り返す。さらに、注目する黒領域を他のものに変えて(ステップS70、No)、ステップS64以降の処理を繰り返す。すべての黒領域を取り出すと処理を終了する。

【0088】このように、図31の線分抽出処理においては、まず矩形内部を適当な長さの重複がある縦縦冊に

分割し、1つの短冊内部で一定の黒画素占有率を満たす部分を抽出して部分線分矩形(黒領域)で表現し、それらを保存する。ここまでは、図11に示した線分の抽出方法と同じである。このとき、保存された部分線分矩形は、下線の一部である高さの小さい矩形の場合もあるが、文字が潰れてそれが下線と接触しているときには、図32のワイルドカードのような高さの大きな矩形の場合もある。これらを横方向に走査していき、全体的な1つの長い線分矩形として抽出する。図32においては、文字列矩形内のワイルドカードの高さは無視されて、他の部分線分矩形と統合され、文字列矩形の下端部分に横長の線分矩形が抽出されている。

【0089】図34、35、36は、線分抽出処理のプログラムコードの例を示している。図35は、図34のC1の位置の α に相当する部分を示しており、図36は、図34のC2の位置の β に相当する部分を示している。また、図37、38、39は、図34、35、36の処理の概要を示すフローチャートである。この処理においては、文字が潰れてできた大きな黒画素塊をワイルドカード矩形として扱い、その前後に8連結で接続される横長の矩形に注目する。そして、ワイルドカード矩形を挟んでお互いに8連結の関係にある矩形を統合していき、1つの横に長い矩形を線分候補の野線として求める。以下、図37、38、39を参照しながら、具体的な処理を説明する。

【0090】図37において処理が開始されると、プロセス14は、まず各部分線分矩形の高さを調べる(ステップS71)。そして、それが文字列矩形の高さ $\times 0.3$ 以上であれば、ワイルドカード矩形としてマークする(ステップS72)。このとき、その部分線分矩形の識別変数useを9とおくことにより、ワイルドカード印をつける。それ以外の部分線分矩形は普通の矩形(スタンダード矩形)として、 $use=0$ とおく(ステップS73)。そして、すべての部分線分矩形をマークしたかどうかを判定し(ステップS74)、まだ部分線分矩形が残っていれば、ステップS71以降の処理を繰り返す。

【0091】すべての部分線分矩形をマークし終わると、1つの矩形をカレント矩形iとして取り出し、 $x1f$ =カレント矩形iの左端座標、 xr =カレント矩形iの右端座標、 yup =カレント矩形iの上端座標、 ybl =カレント矩形iの下端座標、 $line_start y=yup$ 、 $line_end y=ybl$ とおく(ステップS75)。そして、カレント矩形iのuseが0または9であるかどうか調べる(ステップS76)。

【0092】カレント矩形iのuseが0または9であれば、次に $use=0$ かどうかを判定する(ステップS77)。 $use=0$ であれば、 $standard_st=yup$ 、 $standard_en=ybl$ 、 $b_use=0$ 、 $use=1$ 、 $height=ybl-yup+$

1とおく(ステップS78)。b_use=0は、カレント矩形iがワイルドカードではなく、スタンダードとして設定されていることを意味し、use=1はカレント矩形iが使用済みであることを意味する。ステップS76でuse=0でなければ、standard_st=0、standard_en=0、b_use=9、height2=ybl-yup+1とおく(ステップS79)。b_use=9は、カレント矩形iがワイルドカードであるため、スタンダードとして設定されないことを意味する。

【0093】次に、他の部分線分矩形をカレント矩形kとして取り出し、rxlf=カレント矩形kの左端座標、rxr=カレント矩形kの右端座標、ryup=カレント矩形kの上端座標、rybl=カレント矩形kの下端座標とおく(図38、ステップS80)。そして、カレント矩形iがスタンダードとして設定されているかどうか、すなわち、b_use=0であるかどうかを調べる(ステップS81)。b_use=0であれば、次に、カレント矩形kのuseが9であるかどうかを調べる(ステップS82)。ここで、use=9の場合は、カレント矩形iがスタンダードで、カレント矩形kがワイルドカードであることを意味する。

【0094】use=9のとき、 $xr+1 \geq rxlf$ 、 $xr < rxr$ 、 $ybl+1 \geq ryup$ 、および $yup-1 \leq rybl$ が成り立つかどうかを判定する(ステップS83)。これらが成り立つ時、カレント矩形kがカレント矩形iの右側にあり、両者が横と縦に1画素(1ドット)以上の重なりを有することを意味する。これらの条件が成り立つ時、 $xr=rxr$ とおいて、カレント矩形iの右端をカレント矩形kの右端まで延長する(ステップS84)。

【0095】ステップS82でuse=9でないとき、次に、use=0であるかどうかを調べる(ステップS85)。ここで、use=0の場合は、カレント矩形iがスタンダードで、カレント矩形kがワイルドカードでないことを意味する。use=0のとき、 $xr+1 \geq rxlf$ 、 $xr < rxr$ 、 $ybl+1 \geq ryup$ 、および $yup-1 \leq rybl$ が成り立ち、かつ、カレント矩形kの高さが一定範囲内かどうかを判定する(ステップS86)。

【0096】これらの条件が成り立つ時、 $xr=rxr$ 、 $yup=ryup$ 、 $ybl=rybl$ 、 $use=2$ 、 $height=rybl-ryup+1$ とおく(ステップS87)。これは、カレント矩形iの右端をカレント矩形kの右端まで延長し、上下端の座標をカレント矩形kのものに置き換えることを意味する。ここで、use=2はカレント矩形kが使用済みであることを意味する。次に、 $ryup < line_starty$ が成り立つかどうかを判定し(ステップS88)、成り立てば $line_starty=ryup$ とおく(ステップS

89)。さらに、 $rybl > line_endy$ が成り立つかどうかを判定し(ステップS90)、成り立てば $line_endy=rybl$ とおく(ステップS91)。

【0097】これらの処理の後、次にb_use=9かどうかを判定する(図39、ステップS92)。ステップS81でb_use=0でないとき、あるいはステップS83、S85、S86、S88、S90で判定結果がNoのときは、直ちにステップS92以降の処理に移る。

【0098】b_use=9であれば、次に、カレント矩形kのuseが9であるかどうかを調べる(ステップS93)。ここで、use=9の場合は、カレント矩形iとカレント矩形kの両方がワイルドカードであることを意味する。use=9であれば、 $xr+1 \geq rxlf$ および $xr < rxr$ が成り立つかどうかを判定する(ステップS94)。これらが成り立つ時、カレント矩形kがカレント矩形iの右側にあり、両者が横と縦に1ドット以上の重なりを有するので、 $xr=rxr$ とおいて、カレント矩形iの右端をカレント矩形kの右端まで延長する(ステップS95)。

【0099】ステップS93でuse=9でないとき、次に、use=0であるかどうかを調べる(ステップS96)。ここで、use=0の場合は、カレント矩形iがワイルドカードで、カレント矩形kがワイルドカードでないことを意味する。use=0のとき、 $xr+1 \geq rxlf$ および $xr < rxr$ が成り立つかどうかを判定する(ステップS97)。これらの条件が成り立つ時、 $xr=rxr$ 、 $yup=ryup$ 、 $ybl=rybl$ 、 $use=2$ 、 $line_starty=ryup$ 、 $line_endy=rybl$ 、 $height=rybl-ryup+1$ 、 $standard_st=ryup$ 、 $standard_en=rybl$ とおく(ステップS98)。これは、カレント矩形iの右端をカレント矩形kの右端まで延長し、上下端の座標をカレント矩形kのものに置き換えることを意味する。また、use=2はカレント矩形kが使用済みであることを意味する。

【0100】次に、カレント矩形kとしてすべての部分線分矩形を取り出したかどうかを判定する(ステップS99)。ステップS92でb_use=9でないとき、あるいはステップS94、S96、S97で判定結果がNoのときは、直ちにステップS99以降の処理に移る。ステップS99で、残っている部分線分矩形があればステップS80以降の処理を繰り返す。

【0101】すべての部分線分矩形について処理が終われば、b_use=9であるかどうかを判定し(ステップS100)、b_use=9であれば、 $height=height2$ とおく(ステップS101)。ステップS100でb_use=9となるのは、カレント矩形iとそれに連結するすべての矩形がワイルドカードであ

った場合に相当する。

【0102】次に、カレント矩形*i*としてすべての部分線分矩形を取り出したかどうかを判定し（ステップS102）、残っている部分線分矩形があればステップS75以降の処理を繰り返す。ステップS76でカレント矩形*i*のuseが0または9でない場合は、取り出した部分線分矩形が既に使用済みであることを意味するので、直ちにステップS102の処理に移り、次の部分線分矩形を取り出す。

【0103】すべての部分線分矩形について処理が終われば、xlf、xr、line_starty、line_endyを、それぞれ抽出した線分矩形の左端、右端、上端、下端の座標としてファイルyokolineに保存し（ステップS103）、処理を終了する。ここで、yokolineは、1つの文字列矩形から抽出された1つ以上の線分矩形を格納するメモリ領域に対応する。

【0104】図24のステップS50では、以上のようにして文字列矩形から線分が抽出され、さらにそれが下線矩形に相当すれば、その文字列矩形に下線フラグが立てられる。こうして文字列矩形加工処理が終了すると、プロセッサ14は、次に図3のステップS8～S10の処理に相当するタイトル・宛先・発信元抽出処理を行う。図40は、タイトル・宛先・発信元抽出処理のフローチャートである。

【0105】図40において処理が開始されると、まず文字列矩形の相対的な位置、高さ、枠/下線情報を使って、タイトルらしさのポイント計算を行う（ステップS111）。各文字列矩形に対するタイトルらしさのポイント付与の方針は、概ね次の通りである。

(m) プラスポイント

文字列の属性（枠内、下線有り）：高得点

文字列のサイズ（高さ、幅）：大きさに依存する得点

文字列の形（縦横比）：一定以上であれば得点

文字列の相互位置関係（上下間隔、左の矩形の有無）：孤立性が高いほど高得点

文書内の位置（中央、上など）：中央、上は高得点、上下の位置の違いには相対的に少ない得点差

(n) マイナスポイント

文字列の属性（文字列矩形内が1つの文字矩形からなる）：大減点

文字列の相互位置関係（上下近接、重複、上の矩形と左揃い、上の矩形がオーバーラップ）：大減点

文書内の位置（右側にある）：大減点

これらの方針に従い、各文字列矩形に例えば以下の条件で得点を与える。

(o) 野線矩形は得点0

(p) 高さが str_freqの1/2未満は得点0

(q) 幅/高さの比が3未満は得点0

(r) 横幅が str_freqの4倍未満は得点0

(s) (o)、(p)、(q)、(r)の条件に該当す

る文字列矩形以外のものについて、以下の条件で得点を与える。

【0106】[#1] 縦横比：幅/高さの比が3の時、20点

[#2] 上下近接：互いに重複している場合を除き、ある文字列矩形と、上下に隣接する2つの文字列矩形との間隔が両方とも str_freq/2以下の時、-40点

[#3] 片方近接：上または下の文字列矩形だけが16ドットより近接している場合、-20点

[#4] 上下間隔：上下の文字列矩形との間隔が str_freqより大きい場合、20点

[#5] 重複：他の文字列矩形と重複がある場合、-40点

[#6] 中心：文字列矩形の横方向（x方向）の中心座標が、（文書領域の中心座標）±（文書領域幅の40%）以内に入っている場合、30点

[#7] 右側：文字列矩形の中心座標が、文書領域の左から60%の位置より右にあり、かつ、（文書領域の中心座標-文字列矩形の左端座標）が文書領域幅の1/6以下の場合、30点

[#8] 高さ1：文字列矩形の高さが str_freqの0.5倍から1.5倍の間にある場合、20点

[#9] 高さ2：文字列矩形の高さが str_freqの1.5倍と3倍の間の場合30点

[#10] 高さ3：文字列矩形の高さが str_freqの3倍より大きい場合、40点

[#11] 高さ4：文字列矩形の高さが str_freqの3倍より大きく、かつ、文字列矩形の下座標が文書領域の上から1/3以内に入る場合、10点

[#12] 横幅：文字列矩形の幅が文書領域幅の0.4倍より大きい場合、10点

[#13] 下線：文字列矩形に下線フラグがある場合、30点

[#14] 枠：文字列矩形に枠付きフラグがある場合、最大30点を与え、その横幅に比例して減少させる。

【0107】[#15] 左に矩形が無い：左側に同じような座標の文字列矩形が無い場合、または、左側に str_freqの3倍より小さい文字列矩形がある場合、20点

[#16] y座標：最も上にある文字列矩形が20点、そこから下に向かって1点ずつ減少した得点

[#17] 左端揃い：文字列矩形の上に左端に近い他の文字列矩形があると-30点

[#18] オーバラップ（overlap）：文字列矩形の上に左端および右端に近い他の文字列矩形がある場合、または、上の文字列矩形の方が左端右端とも文書領域の端に近い場合、-30点

[#19] 黒領域：大きな文字列矩形で、その内部が1つの黒画素連結領域で成り立っている場合、-40点

図41は、[#18]のオーバーラップしている文字列矩形の例を示している。図41(a)においては、上の文

字列矩形と下の文字列矩形の左右端が近接しており、図41(b)においては、上の文字列矩形の左右端の方が、下の文字列矩形の左右端より文書領域の端に近い。このような場合、下の文字列矩形はタイトルである可能性が低いと考えられる。

【0108】上記(o)、(p)、(q)、(r)、(s)のポイントを、文字列矩形毎に合計し、メモリ15に保存する。次に、ポイントが高い順にタイトル候補として抽出し、結果を保存する(ステップS112)。ここでは、line3内の全文字列矩形を対象にして、それらをポイントが高い順に並び替え、その結果をファイルtitleに格納する。title内には、タイトル候補の第1位の文字列矩形から順に、全文字列矩形が格納される。これにより、第1候補の文字列矩形がタイトル矩形として抽出される。次に、タイトル候補の第1位の文字列矩形から見た相対的位置関係の情報を使って宛先の文字列矩形(宛先矩形)を抽出し、保存する(ステップS113)。また、その相対的位置関係の情報または発信元情報の文字列矩形(発信元情報矩形)を抽出し、それを保存して(ステップS114)、処理を終了する。発信元情報には、文書の発信日、発信者名、レポート番号等が含まれる。

【0109】ステップS113においては、まずタイトルの第1候補の文字列矩形のy方向の位置を求めて、それが最も上であった場合は第1の宛先抽出処理を行い、それ以外の場合は第2の宛先抽出処理を行う。図42は、第1の宛先抽出処理のフローチャートであり、図43は、第2の宛先抽出処理のフローチャートである。

【0110】まず、第1の宛先抽出処理について説明する。図42において処理が開始されると、プロセッサ14は、まずタイトル矩形より下にある文字列矩形の中からキー宛先矩形を抽出し、それを保存する(ステップS121)。ここでは、タイトル矩形より下にあり、高さがst_hの0.6倍からen_hの1.4倍の間にある文字列矩形であって、そのx方向の中心座標がタイトル矩形の中心座標よりも左にあり、幅/高さの比が3より大きいものを、キー宛先矩形として抽出する。そして、このキー宛先矩形より上にある文字列矩形の中に、x方向の中心座標がタイトル矩形の中心座標よりも右にあるような、発信元情報と思える文字列矩形がない場合に、抽出したキー宛先矩形をファイルtoに保存する。

【0111】次に、キー宛先矩形の右にある文字列矩形を宛先矩形として追加する(ステップS122)。ここでは、キー宛先矩形の右にあり、そのy座標が(キー宛先矩形のy座標)±(高さの0.2倍)の範囲内に収まっている文字列矩形を宛先矩形とみなし、キー宛先矩形との重複登録を避けて、toに登録する。

【0112】次に、上下に宛先矩形がある文字列矩形を宛先矩形として追加する(ステップS123)。ここで

は、これまで抽出したto内の宛先矩形の高さの平均値(平均高さ)を求める。そして、タイトル矩形より下の全文字列矩形の内、これまで抽出された宛先矩形でなく、上または下が宛先矩形で、左端の座標が上または下の宛先矩形の左端の座標と一定誤差以内で一致し、かつ、高さが平均高さの2倍未満か、上または下の宛先矩形までの距離が平均高さの1/2未満のものを、宛先矩形としてtoに追加登録する。このような処理を宛先矩形数が増えなくなるまで繰り返す。

【0113】こうして、第1の宛先抽出処理が終了し、to内の文字列矩形が宛先矩形として抽出される。次に、第2の宛先抽出処理について説明する。図43において処理が開始されると、プロセッサ14は、まずタイトル矩形より上にある文字列矩形の中からキー宛先矩形を抽出し、それを保存する(ステップS131)。ここでは、タイトル矩形より上にあり、高さがst_hの0.6倍からen_hの1.4倍の間にある文字列矩形であって、そのx方向の中心座標がタイトル矩形の中心座標よりも左にあり、幅/高さの比が3より大きいものを、キー宛先矩形として抽出する。そして、抽出したキー宛先矩形をファイルtoに保存する。

【0114】次に、キー宛先矩形の右にある文字列矩形を宛先矩形として追加する(ステップS132)。ここでは、キー宛先矩形の右一定距離以内にあり、そのy座標が(キー宛先矩形のy座標)±(高さの0.2倍)の範囲内に収まっている文字列矩形を宛先矩形とみなし、キー宛先矩形との重複登録を避けて、toに登録する。

【0115】次に、上下に宛先矩形がある文字列矩形を宛先矩形として追加する(ステップS133)。ここでは、これまで抽出したto内の宛先矩形の平均高さを求める。そして、タイトル矩形より下の全文字列矩形の内、これまで抽出された宛先矩形でなく、上または下が宛先矩形で、左端の座標が上または下の宛先矩形の左端の座標と一定誤差以内で一致し、かつ、高さが平均高さの2倍未満か、上または下の宛先矩形までの距離が平均高さの1/2未満のものを、宛先矩形としてtoに追加登録する。このような処理を宛先矩形数が増えなくなるまで繰り返す。

【0116】こうして、第2の宛先抽出処理が終了し、to内の文字列矩形が宛先矩形として抽出される。図40のステップS114では、タイトル矩形のy方向の位置を求めて、それが最も上であった場合は、第1の発信元情報抽出処理を行い、それ以外の場合は第2の発信元情報抽出処理を行う。

【0117】第1の発信元情報抽出処理においては、タイトル矩形より下の文字列矩形であって、宛先矩形でないものを対象にして、高さがst_hの0.6倍からen_hの1.4倍の間にあり、かつ、x方向の中心座標がタイトル矩形のそれよりも右にあるものを、宛先矩形として抽出し、ファイルfromに保存する。また、第2の発信

元情報抽出処理においては、タイトル矩形より上の文字列矩形であって、宛先矩形でないものを対象にして、第1の発信元情報抽出処理と同様の文字列矩形を宛先矩形として抽出し、ファイルfromに保存する。こうして、from内の文字列矩形が発信元情報矩形として抽出される。

【0118】第1および第2の発信元情報抽出処理は、第1および第2の宛先抽出処理に比べて簡単になっているが、宛先抽出処理と同様に、一定の条件を満たす他の文字列矩形をさらに発信元情報矩形に加えるようにしてもよい。

【0119】図44は、タイトルと宛先／発信元情報の第1の配置を示している。図44においては、タイトル矩形が最も上にあるため、第1の宛先抽出処理および第1の発信元情報抽出処理が適用される。図45、46、47は、それぞれタイトルと宛先／発信元情報の第2、第3、第4の配置を示している。これらの配置においては、タイトル矩形が最も上ではないので、第2の宛先抽出処理および第2の発信元情報抽出処理が適用される。また、図48は、複数の宛先／発信元情報の例を示している。図48においても、第2の宛先抽出処理および第2の発信元情報抽出処理が適用される。

【0120】図45、47、48のような配置の場合は、第2の発信元情報抽出処理を行うと、タイトル矩形より下にある発信元情報矩形が抽出されない。そこで、タイトル矩形が最も上にない場合でも、第1の発信元情報抽出処理を行う構成としてもよい。また、第1および第2の発信元情報抽出処理を併用してもよい。

【0121】図49は、タイトル・宛先・発信元抽出処理により生成されたファイルtitle、to、fromの内容を示している。図49においては、文字列矩形「ソフトウェア販推レポート 送付表」がタイトル矩形として抽出され、それに続く左揃いの文字列矩形が複数の宛先矩形として抽出されている。また、右下の数字が発信元情報として抽出されている。

【0122】図50は、タイトル・宛先・発信元抽出処理による他の抽出結果を示している。図50においては、文字列矩形「外部発表の受付状況について（送付）」がタイトル矩形として抽出され、その左上にある文字列矩形が宛先矩形として抽出されている。また、タイトル矩形の右上の複数の文字列矩形が発信元情報として抽出されている。

【0123】こうして、抽出されたタイトル矩形、宛先矩形、および発信元情報矩形は、図3のステップS11の認識処理により文字列として認識される。このとき、各認識対象の矩形から1文字ずつ文字が切り出され、各文字毎に文字認識が行われる。そして、認識結果は、例えば、電子ファイリング装置11内の画像ファイルのキーワードとして用いられる。

【0124】以上の実施形態において、図31の線分抽出処理は、図24のステップS50の下線抽出処理のみ

ならず、図3のステップS6で大きな矩形から横線分を抽出する際にも適用できる。これにより、大きな矩形内のワイルドカードの高さを無視して横方向の線分矩形を抽出し、それを一部分とする枠線を識別することができる。

【0125】ところで、図3から図50までで説明した実施形態では、表の外部にある領域からタイトルを抽出する技術について記述されている。表の内部にタイトルがある場合には、図5のステップS24で表矩形が処理対象から除外されているため、表内のタイトルを抽出することができない。

【0126】一般に表を含む文書においては、その表の外に文書全体のタイトルがあることが多いが、会社内の文書の中には、定型の事務文書等のように表の内部にタイトルがあるものもある。また、表の外にタイトルがあっても、それが「議事録」のような一般的な文書名で、電子ファイリングシステムの検索時に必要な文書を特定できるキーワードとなるタイトルは表内の1つの欄内に記されていることもある。

【0127】このような場合に、文字認識などの処理時間のかかる技術を使わずに、表内の有効なタイトル部分を高速に抽出することが望まれる。以下では、表を含む一般文書の文書画像から、表内にある「表題」や「会社名」のようなタイトルらしい欄の名称を表現する部分（項目部分）と、項目の具体的な内容を表すタイトル部分とを抽出する実施形態を説明する。

【0128】図51は、表形式の社内文書の例を示している。図51の表形式文書では、表罫線で囲まれた表内の左上にある「表題」が項目部分に相当し、その右にある「マルチメディアとパターン認識シンポジウム」がタイトル部分に相当する。このように、横書き文書の場合には、表内のタイトル部分は、通常、項目部分の右側にあると考えてよい。

【0129】図52は、図2のタイトル抽出システムによる表内タイトル抽出処理のフローチャートを示している。図52の処理においては、前提条件として横書き文書を対象としているが、図3の処理と同様に、縦書き文書にも対応可能である。

【0130】図52において処理が開始されると、光電変換装置12が文書を読み取り、文書画像としてメモリ15に格納する（ステップS141）。ここでも、図3のステップS1と同様にして、原画像を圧縮画像に変換して保存する。図51の文書から作成された圧縮画像は図53のようになる。

【0131】次に、プロセッサ14が、文書画像にラベリング処理を施し、矩形高さの最頻値を求めて、それをもとに大きな矩形を抽出する（ステップS142）。ここでの処理は、図5のステップS21、S22、S23の処理と同様である。ただし、枠矩形の抽出は行わず、ファイルboxに保存される矩形は閾値th_large

より大きな矩形である。図53の文書画像のラベリング結果は図54のようになる。

【0132】次に、抽出された大きな矩形から表を囲む矩形(表矩形)を抽出し(ステップS143)、表矩形の中からタイトルを含むものを選択する(ステップS144)。ここでは、例えば最も面積の大きな表矩形が選択され、以下の処理は選択された表矩形の内部を対象にして行われる。

【0133】プロセッサ14は、まず表矩形の内部から文字列(あるいは行)を抽出し、文字列の外接矩形(文字列矩形)を求め、その座標をメモリ15に保存する(ステップS145)。次に、保存した文字列矩形から、横幅が小さい矩形や縦長矩形をノイズ矩形として除去し(ステップS146)、2つ以上の文字列矩形を統合する(ステップS147)。

【0134】ステップS145の処理は、基本的に図5のステップS25からS31までの処理と同様である。また、ステップS146の処理は、図24のステップS41の処理と同様であり、ステップS147の処理は、ステップS42からS44までの処理と同様である。

【0135】ここまでの処理で、表内から抽出された文字列矩形が整理されるが、これらの文字列矩形は表罫線の一部を含んでいる場合もあり得る。そこで、文字列矩形の中の罫線部分を抽出し、その部分を境にして文字列矩形を分割する(ステップS148)。

【0136】次に、タイトルに相当する文字列矩形を抽出するために、文字列矩形内の文字数を計算する(ステップS149)。ここで計算された文字数は、文字列矩形の属性としてステップS152の処理で用いられる。

【0137】ステップS148の処理により表罫線で囲まれた欄毎の文字列矩形が抽出されるが、元の表の外形が矩形ではない場合には、表の外にある文字列矩形が残されている可能性がある。そこで、上罫線のチェックを行って(ステップS150)、上側に表罫線がないような文字列矩形は表外の文字列矩形とみなし、それを除去する。

【0138】次に、表内の文字列矩形を表矩形の左上座標に近い順に並び替える(ステップS151)。そして、文字列矩形の文字数が一定の条件を満たす場合に、その文字列矩形を項目部分またはタイトル部分として抽出して(ステップS152)、処理を終了する。このとき、条件を満たす文字列矩形を、表矩形の左上に近いものから優先的にタイトル矩形の候補とする。

【0139】次に、表内タイトル抽出処理の各ステップで行われる具体的な処理内容を説明する。図55は、図52のステップS143の表矩形抽出処理のフローチャートである。この表矩形抽出処理に先立ってステップS142の処理を行っておくことで、処理対象が一定以上大きな矩形に限られるため、表矩形の抽出が効率化される。

【0140】図55において処理が開始されると、プロセッサ14は、まずbox内の大きな矩形から高さが閾値より大きなものを抽出する(ステップS161)。ここでは、例えば矩形高さの最頻値freqの5倍より大きい(高い)矩形が抽出され、表矩形としてファイルlarge_4baiに格納される。ステップS161で抽出された表矩形は、ステップS150の上罫線チェックの際に用いられる。

【0141】次に、box内の大きな矩形から横幅が閾値より大きなものを抽出して(ステップS162)、処理を終了する。ここでは、例えば横幅が文書画像の横幅の0.7倍より大きな矩形が抽出され、表矩形としてファイルlargewideに格納される。

【0142】図52のステップS144では、ステップS162で抽出されたいくつかの表矩形のうちで最も大きなものが選択される。ここでは、例えば、largewide内の複数の矩形からその面積が最大のものが選択されて、処理対象となる。図54の文書画像の場合は、largewide内に格納される大きな矩形は表矩形80のみであるため、自動的にこれが処理対象の表矩形となる。

【0143】次に、図52のステップS145では、選択された表矩形内部の文字矩形を対象にして文字列矩形の抽出が行われる。しかし、次のいずれかの条件に該当する矩形は処理対象から除外する。

(t) 枠矩形

(u) 高さがfreqの3倍より大きく、高さ／幅の比が0.4より小さな横長矩形(v) 文書画像全体の高さの1/3より大きな矩形

このうち、(t)の枠矩形は、図5のステップS23と同様の処理により抽出することができる。

【0144】ステップS145、S146、S147の処理を行った後に得られる統合された文字列矩形は、図56のようになる。図56において、例えば文字列矩形81、82、83等は、表罫線により仕切られた本来別々の複数の文字列を含んでいる。そこで、表内の文字列を正しく抽出するために、ステップS148で文字矩形間の縦罫線を境界にして文字列矩形を分割する。以下、図57から図65までを参照しながら、この文字列分割処理について説明する。

【0145】文字列分割方法としては、大きく分けて2つの方法が考えられる。図57は、第1の文字列分割処理のフローチャートである。第1の文字列分割処理においては、プロセッサ14は、各文字列矩形に含まれる任意の2つの隣接文字矩形の間に縦罫線があるかどうかをチェックする。このとき、まず文字列矩形内に含まれる文字矩形を横方向に並び替え、それらの間に黒画素があるかどうかチェックする。黒画素がある場合はその位置で文字列矩形を分割して、複数の新しい文字列矩形を生成する。

【0146】図57において処理が開始されると、プロ

セッサ14は、まず文字列矩形内の文字矩形をx座標（横座標）の小さいものから順にソートする（ステップS171）。ステップS147までの処理においては、文字列矩形内の文字矩形は一般にy座標（縦座標）の小さい順にソートされており、横方向の順序が反映されていない。そこで、実際の文字の並びに対応するように、文字矩形の記憶順序が変更される。

【0147】例えば、図58に示す文字列矩形91の場合、文字列分割処理の前には文字矩形92、95、93、94の順にソートされて、記憶されている。これらの文字矩形をx座標でソートし直すことにより、図59に示すように文字矩形92、93、94、95の順に正しく記憶される。

【0148】次に、文字列矩形の左端のx座標、右端のx座標、上端のy座標、下端のy座標を、それぞれ $sx1$ 、 $sx2$ 、 $sy1$ 、 $sy2$ とおき（ステップS172）、文字列矩形内の最も左の文字矩形に注目し、それをカレント矩形とする（ステップS173）。そして、カレント矩形の上端のy座標、下端のy座標、右端のx座標を、それぞれ cyl 、 $cy2$ 、 $cx2$ とおき（ステップS174）、カレント矩形の右にある文字矩形の上端のy座標、下端のy座標、左端のx座標を、それぞれ $ry1$ 、 $ry2$ 、 $rx1$ とおく（ステップS175）。

【0149】次に、直線 $x=cx2$ 、 $x=rx1$ 、 $y=\max(cy1, ry1)$ 、 $y=\min(cy2, ry2)$ で囲まれた矩形領域内に黒画素があるかどうかをチェックする（ステップS176）。ここで、この矩形領域は、カレント矩形とカレント矩形の右の文字矩形の間に位置する領域である。

【0150】上記矩形領域内に黒画素があれば、そこに縦罫線があるとみなして、座標 $x=sx1$ 、 $cx2$ 、 $y=sy1$ 、 $sy2$ で表される矩形を文字列矩形として登録し、 $sx1=rx1$ とする（ステップS177）。

【0151】次に、カレント矩形の右の文字矩形が文字列矩形の中で最も右にあるかどうかを調べ（ステップS178）、そうでない場合はカレント矩形の右の文字矩形を新たにカレント矩形として（ステップS179）、ステップS174以降の処理を繰り返す。ステップS176で上記矩形領域内に黒画素がなければ、そのままステップS178以降の処理を行う。

【0152】そして、ステップS178においてカレント矩形の右の文字矩形が最も右の矩形である場合は、座標 $x=sx1$ 、 $sx2$ 、 $y=sy1$ 、 $sy2$ で表される矩形を文字列矩形として登録して（ステップS180）、処理を終了する。

【0153】このような第1の文字列分割処理によれば、カレント矩形とカレント矩形の右の矩形の間に縦罫線が検出される度に、その左側の1つ以上の文字矩形が文字列矩形として登録される。したがって、元の文字列矩形に縦罫線が2本以上含まれていても、必ずそれらの

位置で文字列矩形が分割される。

【0154】例えば、図60のような表内の文字列矩形101の場合、文字矩形102、103、104、105、106、107を含んでおり、文字矩形102と文字矩形103の間には表の縦罫線が通っている。この文字列矩形101を対象に第1の文字列分割処理を行うと、文字矩形102がカレント矩形のとき、文字矩形102と文字矩形103の間の領域に黒画素が検出される（ステップS176、Yes）。そこで、図61に示すように、文字矩形102を含む矩形が文字列矩形108として登録される（ステップS177）。

【0155】その後、文字矩形103が新たにカレント矩形となって（ステップS179）、同様の処理が繰り返されるが、縦罫線は検出されない。そして、文字矩形106がカレント矩形となったとき、文字矩形103、104、105、106、107を含む矩形が文字列矩形109として登録され（ステップS180）、処理が終了する。こうして、元の文字列矩形101は、文字列矩形108と109に分割される。

【0156】図62および図63は、第2の文字列分割処理のフローチャートである。第2の文字列分割処理においては、プロセッサ14は、各文字列矩形の内部を対象にして再度ラベリング処理を施す。このとき、まず文字列矩形を構成する各文字矩形の座標を記憶しておき、それとは別に、文字列矩形内のラベリング処理により得られた文字矩形の座標を獲得する。

【0157】縦罫線の一部が文字列矩形内にあるとすると、前者の文字矩形群と後者の文字矩形群とを比較した場合、後者の方が縦罫線の分だけ矩形の数が増えるため、両者の間に差異が生じるはずである。そこで、前者と比較して後者に余分な文字矩形が出現した位置で文字列矩形を分割する。

【0158】例えば、図60の文字列矩形101の場合、その内部にラベリング処理を施して得られる文字矩形は図64のようになる。図60の文字矩形群と図64の文字矩形群とを比較すると、図64の方が余分な矩形110を含んでいることが分かる。この矩形110は文字列矩形101内に含まれた縦罫線に相当し、この位置で文字列矩形101を分割することができることを表している。

【0159】図62において処理が開始されると、プロセッサ14は、まず文字列矩形内の文字矩形の集合をOとし（ステップS181）、文字列矩形内をラベリング処理して求めた文字矩形の集合をNとする（ステップS182）。そして、集合OとN内の文字矩形をそれぞれx座標でソートし（ステップS183）、文字列矩形の左端のx座標、右端のx座標、上端のy座標、下端のy座標を、それぞれ $sx1$ 、 $sx2$ 、 $sy1$ 、 $sy2$ とおく（ステップS184）。x座標によるソート処理は、図57のステップS172と同様に行う。

【0160】次に、登録フラグ=0とおき、O内の最も左の文字矩形を矩形OOとし、N内の最も左の文字矩形を矩形NNとする。そして、 $x2=OO$ の右端のx座標、 $prev=x2$ とおく（ステップS185）。以後、登録フラグは0または1の値をとる。

【0161】次に、OOとNNの左上頂点および右下頂点の座標が一致するかどうかをチェックする（ステップS186）。これらがともに一致すればOOとNNは同じ矩形であるとみなし、次に、登録フラグが1かどうかを判定する（ステップS187）。

【0162】登録フラグが0の場合は、OOの右の矩形を新たにOOとおき、NNの右の矩形を新たにNNとおく（ステップS188）。そして、 $prev=x2$ とおいた後（ステップS189）、 $x2=OO$ の右端のx座標とおき（ステップS190）、OOが文字列矩形の中で最も右の文字矩形かどうかを判定する（ステップS191）。そして、OOの右にまだ文字矩形があれば、ステップS186以降の処理を繰り返す。

【0163】ステップS186において、OOとNNの座標が一致しない場合はNNが縦罫線に相当するとみなし、次に登録フラグが0かどうかを判定する（図63、ステップS195）。そして、登録フラグが0であれば、座標 $x=sx1$ 、 $prev$ 、 $y=sy1$ 、 $sy2$ で表される矩形を文字列矩形として登録し（ステップS196）、登録フラグ=1とおく（ステップS197）。これにより、OOの左の文字矩形を含む矩形が文字列矩形として登録される。

【0164】次に、縦罫線とみなされたNNの右の矩形を新たにNNにおいて（ステップS198）、ステップS186以降の処理を繰り返す。ステップS195において登録フラグが0でなければ、そのままステップS198以降の処理を行う。

【0165】ステップS187において、登録フラグが1の場合はOOを新たな文字列の先頭文字とみなして、 $x2=OO$ の右端のx座標、 $sx1=OO$ の左端のx座標とおく（ステップS192）。そして、 $prev=x2$ 、登録フラグ=0とおき（ステップS193、S194）、ステップS191以降の処理を行う。

【0166】そして、ステップS191においてOOが最も右の文字矩形の場合は、座標 $x=sx1$ 、 $x2$ 、 $y=sy1$ 、 $sy2$ で表される矩形を文字列矩形として登録して（ステップS199）、処理を終了する。

【0167】このような第2の文字列分割処理によれば、集合N内にあってO内にはない余分な矩形が検出される度に、その左側の1つ以上の文字矩形が文字列矩形として登録される。また、その後はO内の次の矩形が文字列の左端に設定されるので、余分な縦罫線は文字列矩形から除去される。

【0168】例えば、図64の文字列矩形101の場合、集合Oは文字矩形102、103、104、10

5、106、107からなり、集合Nは文字矩形102、110、103、104、105、106、107からなる。そして、OOが文字矩形103でNNが文字矩形110のとき、文字矩形110が縦罫線とみなされる（ステップS186、No）。そこで、図61に示すように、文字矩形102を含む矩形が文字列矩形108として登録される（ステップS196）。

【0169】その後、文字矩形103が新たにNNとなって（ステップS198）、同様の処理が繰り返されるが、縦罫線に相当する矩形は検出されない。そして、文字矩形107がOOとなったとき、文字矩形103、104、105、106、107を含む矩形が文字列矩形109として登録され（ステップS199）、処理が終了する。こうして、元の文字列矩形101は、第1の文字列分割処理の結果と同様に、文字列矩形108と109に分割される。

【0170】第1および第2の文字列分割処理を比較すると、それらの機能は基本的に同じであるが、第1の文字列分割処理の方が処理速度が速いという利点がある。図56の文字列矩形に文字列分割処理を施した結果は図65のようになる。図56と図65とを比較すると、元の文字列矩形81は文字列矩形111、112、および113に分割されていることが分かる。また、文字列矩形82は文字列矩形114と115に分割され、文字列矩形83は文字列矩形116と117に分割されている。

【0171】文字列矩形の分割が終了すると、次に、図52のステップS149において、プロセッサ14は、文字列矩形内の文字矩形の形状からその文字数を計算する。ここでは、文字矩形の高さと幅の比からそれを構成する文字数を抽出する。

【0172】図66は、このときの文字矩形とその文字数の関係を示している。図66において、文字矩形の高さをH、幅をWとすると、一般に1つの文字の高さと幅はほぼ等しいと考えられるので、この文字矩形内にある文字数は $[W/H]$ 個と表すことができる。ここで、 $[W/H]$ は、実数 W/H の小数点以下を切り捨てる演算記号である。

【0173】ステップS148の文字列分割処理により表矩形内の文字列矩形が正しく分割されるが、表矩形内には実際の表の外にある文字列矩形が含まれている可能性がある。図67は、このような表矩形内の表外文字列矩形の例を示している。図67において、太線で示された表罫線の外周は矩形ではないため、その表矩形121内には表外にある文字列矩形122が含まれている。一方、文字列矩形122と同じ行にある文字列矩形123は表内の文字列矩形である。

【0174】図68は、図54の表矩形80内の文字列矩形を示している。図68の文字列矩形のうち、文字列矩形131が表外の文字列矩形に相当する。表内のタイ

トルを抽出するためには、文字列矩形122や131のような表外の文字列矩形を表内の文字列矩形と区別し、表矩形内から取り除く必要がある。

【0175】そこで、ステップS150において、上に他の文字列矩形がない文字列矩形を対象に、その上に罫線があるかどうかをチェックし、罫線がなければその文字列矩形を除去する。

【0176】図69は、このような上罫線チェック処理のフローチャートである。図69において処理が開始されると、プロセッサ14は、まず図24のステップS42と同様の方法で、文字列矩形間の接続関係を表す連結関係表を作成する(ステップS201)。そして、連結関係表を用いて上に他の文字列矩形がない文字列矩形を求め、それらのうち上に罫線がないものを除去して(ステップS202)、処理を終了する。

【0177】図70は、ステップS202の表外文字列矩形除去処理のフローチャートである。図70の表外文字列矩形除去処理においては、表矩形内のすべての文字列矩形の連結関係表を参照して、文字列矩形の上に他の文字列矩形がないものを抽出する。そして、抽出した文字列矩形の上の特定領域内を探索して、黒画素を含むバイト数の合計Mを求める。ただし、8画素=1バイトとする。

【0178】Mが探索範囲の横の長さをバイト数で表したとき値L以上であれば、この範囲に横罫線があるとみなして、その文字列矩形を表内文字列矩形として残す。もし、 $M < L$ となるような文字列矩形があれば、その文字列矩形の上には横罫線がないとみなし、それを表外文字列矩形として除去する。

【0179】図70において処理が開始されると、プロセッサ14は、まず表矩形内の文字列矩形からなる集合を、表内文字列矩形の集合Sとする(ステップS211)。次に、S内で、他のS内の文字列矩形を上矩形とする接続関係を持たないものを抽出し、それらの集合をS1とする(ステップS212)。例えば、図67の場合は、斜線の文字列矩形122と123がS1の要素となる。

【0180】次に、S1内の1つの文字列矩形をSSとし(ステップS213)、SSの左端のx座標、右端のx座標、上端のy座標、下端のy座標を、それぞれsx1、sx2、sy1、sy2とおく(ステップS214)。

【0181】次に、SSの上にある他の表矩形または表矩形外の文字列矩形を求め、その左端のx座標、右端のx座標、上端のy座標、下端のy座標を、それぞれux1、ux2、uy1、uy2とおく(ステップS215)。ここで、他の表矩形としては、図55のステップS161で抽出されてlarge_4baiに格納されている表矩形が参照される。

【0182】次に、直線 $x = \max(sx1, ux$

$1)$, $x = \min(sx2, ux2)$, $y = sy1$, $y = uy2$ で囲まれた矩形領域の横幅のバイト数をLとする(ステップS216)。この矩形領域の横幅は、SSの横幅とその上の矩形の横幅の重複部分の長さに相当する。このとき、Lは次式で与えられる。

$$L = \min(sx2, ux2) / 8 - \max(sx1, ux1) / 8 + 1$$

次に、上記矩形領域内で黒画素を求め、8画素を1バイトとして、そのバイト数の総和Mを計算する(ステップS217)。

【0183】ステップS215で求めたSSの上にある矩形が文字列矩形の場合は、ステップS216のLと、ステップS217における黒画素の探索範囲は、図71のようになる。また、SSの上にある矩形が他の表矩形の場合は、それらは図72のようになる。

【0184】次に、MとLの大きさを比較し(ステップS218)、MがL未満であればSSの上に横罫線がないものとみなして、SSを表外の文字列矩形と判定する。そこで、SSを集合Sから除去する(ステップS219)。

【0185】次に、SSがS1内の最後の文字列矩形かどうかを判定し(ステップS220)、最後でなければステップS213以降の処理を繰り返す。そして、S1内の文字列矩形をすべて処理すると、処理を終了する。

【0186】ステップS215において、SSの上に表矩形も文字列矩形もない場合は、ステップS217で文書画像の上端までの範囲を探索して、黒画素を求めればよい。このときの探索範囲は図73のようになり、その横幅はSSの横幅に一致する。

【0187】図70の処理により、図68の表外文字列矩形131が除去され、残された表内文字列矩形は図74のようになる。こうして得られた表内文字列矩形を対象にして、それらの位置や文字数の関係からタイトルの候補が抽出される。

【0188】図75は、図52のステップS151およびS152で行われるタイトル候補出力処理のフローチャートである。横書き文書の場合は、一般に左上に近い文字列ほどタイトルである可能性が高いので、図75のタイトル候補出力処理においては、まず文字列矩形を表の左上に近い順に並び替える。そして、その順番やステップS149で求めた文字数等の情報を使用して、文字列矩形の表内タイトルらしさの優先順位を決め、その順にタイトル候補として出力する。

【0189】優先順位の付け方としては、大きく分けて次の3通りが考えられる。

(w) 表の左上に近い順に優先順位を付ける。

(x) 隣りの文字列矩形内の文字矩形の文字数を調べ、その関係をもとに優先順位を決める。表内のタイトルには、「題名」や「表題」のようにタイトルであることを示す項目名がタイトルの左(または上)の位置にある場

合が多い。このような項目名とタイトルの関係は、それらの文字数を用いて表すことができる。例えば、2文字から数文字程度の文字列の右側（または下側）に、数文字から十数文字程度の文字列がある場合に、項目名とタイトルのペアがあると判断することができる。そこで、そのようなペアについて、上から順に優先順位を付ける。

(y) 一定の文字数の条件、または隣りの文字列矩形との間の一定の文字数の関係を満足するものだけを対象にして、表の左上に近い順に優先順位を付ける。

【0190】この場合は、表内の文字列矩形を左上に近い順に調べていき、文字列矩形内の文字数の合計が閾値以上であれば、その文字列矩形を項目候補とする。さらに、その文字列矩形の右側に他の文字列矩形があれば、その文字列矩形内の文字数にかかわらず、それをタイトル候補とする。

【0191】これは、元々1つの欄に項目とタイトルが併記されており、「項目：タイトル」のように1つの文字列矩形に両方の要素が含まれる場合を救済するためである。また、文字数の大きな文字列矩形は、それだけで表内タイトルらしいといえる。このような文字列矩形は項目候補として出力された場合でも、文字認識の結果からタイトルらしいと考えられれば、タイトルとして使用することができる。

【0192】文字列矩形内の文字数が閾値未満の場合は、その右側に他の文字列矩形があり、かつ、その中の文字数が閾値以上の場合に、前者を項目候補、後者をタイトル候補とする。

【0193】上記(w)、(x)、(y)の各方法について、20種類の文書画像を用いて実験した結果、

(y)の方法の場合に表内タイトルが候補の上位に入りやすく、これが最も抽出性能が良いことが分かった。そこで、図75の処理では(y)の方法に従って優先順位を決めている。

【0194】図75において処理が開始されると、プロセッサ14は、まず図76に示すような各表内文字列矩形の左上頂点の座標(x1, y1)を用いて、 $x1 + y1$ の値の小さい順に、それらの文字列矩形をソートする(ステップS221)。そして、表内文字列矩形の集合をSとし(ステップS222)、S内で $x1 + y1$ の値が最も小さいものをカレント矩形SSとする(ステップS223)。

【0195】次に、SS内の文字矩形の文字数の合計が閾値TITLEMOJISUU以上かどうかを判定する(ステップS224)。例えば、TITLEMOJISUU=7とする。SSの文字数がTITLEMOJISUU以上であれば、SSの右側に他の文字列矩形があるかどうかを調べる(ステップS225)。右側に文字列矩形がなければ、カレント矩形SSを項目候補として出力し(ステップS226)、SSがS内の最後の文字列矩形かどうかを判定する(ステップ

S227)。最後の文字列矩形でなければ、SSの次に $x1 + y1$ の値の小さな文字列矩形を新たにSSとし(ステップS231)、ステップS224以降の処理を繰り返す。

【0196】ステップS225においてSSの右側に文字列矩形がある場合は、カレント矩形SSを項目候補、その右側の文字列矩形をタイトル候補として出力し(ステップS230)、ステップS227以降の処理を行う。

【0197】また、ステップS224においてSS内の文字数がTITLEMOJISUU未満の場合は、SSの右側に他の文字列矩形があるかどうかを調べる(ステップS228)。右側に文字列矩形があれば、その文字数の合計がTITLEMOJISUU以上かどうかを判定する(ステップS229)。そして、それがTITLEMOJISUU以上であれば、ステップS230以降の処理を行う。

【0198】ステップS228においてSSの右側に文字列矩形がない場合、および、ステップS229において右側の文字列矩形の文字数がTITLEMOJISUU未満の場合は、ステップS227以降の処理を行う。そして、ステップS227においてSSが最後の文字列矩形であれば、処理を終了する。

【0199】このタイトル候補出力処理によれば、次の3つの場合に該当する文字列矩形が項目またはタイトル候補として出力される。

(α) カレント矩形の文字数が閾値以上で、その右側に文字列矩形がない場合、カレント矩形を項目候補として出力する。

(β) カレント矩形の文字数が閾値以上で、その右側に文字列矩形がある場合、カレント矩形を項目候補、右側の文字列矩形をタイトル候補として出力する。

(γ) カレント矩形の文字数が閾値未満で、その右側の文字列矩形の文字数が閾値以上である場合、カレント矩形を項目候補、右側の文字列矩形をタイトル候補として出力する。

【0200】図77は、こうして抽出された表内タイトルの第1候補を示している。図77において、文字列矩形111は項目候補であり、文字列矩形112はタイトル候補である。このような表内タイトル抽出処理によれば、様々な表を含んだ文書画像に対しても、特別な操作や辞書等を用いずに、表内の項目およびタイトルの領域を抽出することができる。

【0201】こうして抽出された項目候補およびタイトル候補の文字列矩形は、図3のステップS11と同様の処理により文字列として認識される。このとき、実際には、項目候補として抽出された文字列がタイトル文字列を含んでいる場合もあるので、認識結果のうち適当な部分を項目名あるいはタイトルとして使用する。

【0202】本発明においては、文字領域および文字列領域の形状は必ずしも矩形でなくてもよく、直線または

曲線により囲まれた任意の形状の領域を用いることができる。

【0203】

【発明の効果】本発明によれば、様々な文書画像に対して、特別な操作を行ったり、辞書等を用意したりしなくても、タイトル、宛先、発信元情報に相当する領域を容易に抽出することができる。これにより、画像データから抽出した文字列等をその画像のキーワードとして用いることもできるようになる。

【図面の簡単な説明】

【図1】本発明の原理図である。

【図2】システム構成図である。

【図3】文書画像のタイトル抽出処理のフローチャートである。

【図4】文書画像データを示す図である。

【図5】文字列抽出処理のフローチャートである。

【図6】ラベリング処理後の外接矩形を示す図である。

【図7】高さのヒストグラムを示す図である。

【図8】高さの最頻値を求めるためのヒストグラムを示す図である。

【図9】矩形高さテーブルを示す図である。

【図10】矩形高さテーブルの内容に対応するヒストグラムを示す図である。

【図11】大きな矩形から抽出された線分矩形を示す図である。

【図12】部分線分矩形を示す図である。

【図13】連結した部分線分矩形を示す図である。

【図14】枠矩形を示す図である。

【図15】重複している外接矩形を示す図である。

【図16】ネストしている外接矩形を示す図である。

【図17】二等辺三角形のヒストグラムを示す図である。

【図18】重複・ネスト除去後の外接矩形を示す図である。

【図19】矩形間の連結関係を示す図である。

【図20】連結関係表を示す図である。

【図21】文字列矩形を示す図である。

【図22】文字列矩形の抽出処理を示す図である。

【図23】抽出された文字列矩形を示す図である。

【図24】文字列矩形加工処理のフローチャートである。

【図25】ノイズ除去後の文字列矩形を示す図である。

【図26】文字列矩形の統合処理を示す図である。

【図27】統合された文字列矩形を示す図である。

【図28】文書領域を示す図である。

【図29】下線矩形を示す図である。

【図30】枠付き・罫線・下線チェック後の文字列矩形を示す図である。

【図31】線分抽出処理のフローチャートである。

【図32】ワイルドカードがある場合の線分矩形を示す

図である。

【図33】ワイルドカードを示す図である。

【図34】線分抽出処理のコードを示す図（その1）である。

【図35】線分抽出処理のコードを示す図（その2）である。

【図36】線分抽出処理のコードを示す図（その3）である。

【図37】線分抽出処理の詳細フローチャート（その1）である。

【図38】線分抽出処理の詳細フローチャート（その2）である。

【図39】線分抽出処理の詳細フローチャート（その3）である。

【図40】タイトル・宛先・発信元抽出処理のフローチャートである。

【図41】オーバーラップしている文字列矩形を示す図である。

【図42】第1の宛先抽出処理のフローチャートである。

【図43】第2の宛先抽出処理のフローチャートである。

【図44】タイトルと宛先／発信元の第1の配置を示す図である。

【図45】タイトルと宛先／発信元の第2の配置を示す図である。

【図46】タイトルと宛先／発信元の第3の配置を示す図である。

【図47】タイトルと宛先／発信元の第4の配置を示す図である。

【図48】複数の宛先／発信元を示す図である。

【図49】タイトルおよび宛先・発信元情報の抽出結果を示す図である。

【図50】タイトルおよび宛先・発信元情報の他の抽出結果を示す図である。

【図51】表形式文書を示す図である。

【図52】表内タイトル抽出処理のフローチャートである。

【図53】表形式文書の画像データを示す図である。

【図54】表形式文書のラベリング結果を示す図である。

【図55】表矩形抽出処理のフローチャートである。

【図56】表形式文書の文字列矩形を示す図である。

【図57】第1の文字列分割処理のフローチャートである。

【図58】文字列矩形内の文字矩形の順位を示す図である。

【図59】ソート後の文字矩形の順位を示す図である。

【図60】縦罫線を含む文字列矩形を示す図である。

【図61】分割された文字列矩形を示す図である。

【図62】第2の文字列分割処理のフローチャート（その1）である。

【図63】第2の文字列分割処理のフローチャート（その2）である。

【図64】文字列矩形内のラベリング結果を示す図である。

【図65】分割処理後の文字列矩形を示す図である。

【図66】文字矩形と文字数の関係を示す図である。

【図67】表矩形内の表外文字列矩形を示す図である。

【図68】表矩形内の文字列矩形を示す図である。

【図69】上罫線チェック処理のフローチャートである。

【図70】表外文字列矩形除去処理のフローチャートである。

【図71】第1の探索範囲を示す図である。

【図72】第2の探索範囲を示す図である。

【図73】第3の探索範囲を示す図である。

【図74】表外文字列矩形除去後の文字列矩形を示す図である。

【図75】タイトル候補出力処理のフローチャートである。

【図76】文字列矩形の左上頂点の座標を示す図である。

【図77】表内タイトルの抽出結果を示す図である。

【符号の説明】

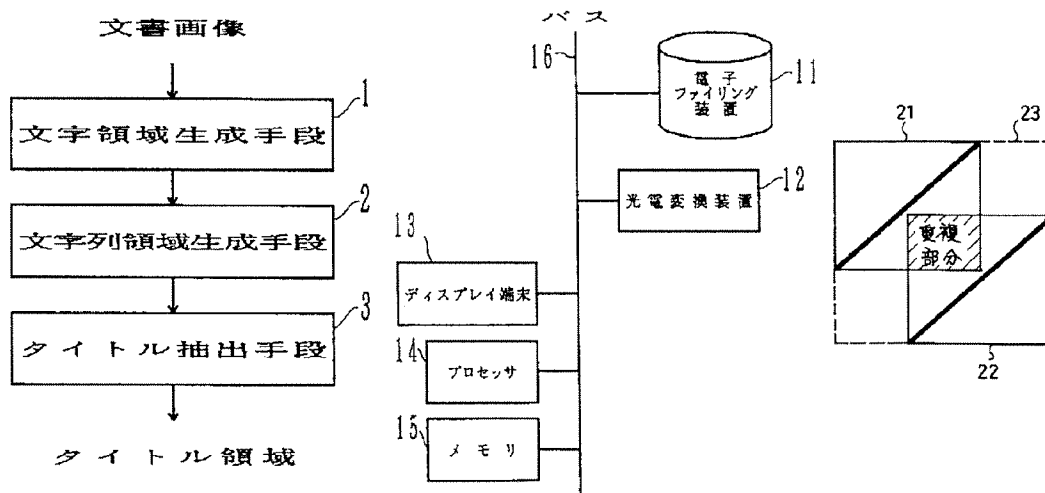
- 1 文字領域生成手段
- 2 文字列領域生成手段
- 3 タイトル抽出手段
- 11 電子ファイリング装置
- 12 光電変換装置
- 13 ディスプレイ端末
- 14 プロセッサ
- 15 メモリ
- 16 バス
- 21、22、23、24、25、26、27、31、32、33、34、35、61、62、63、64、65 外接矩形
- 36、37、38、39、40 二等辺三角形
- 41、42、43 二等辺三角形のヒストグラム
- 51、52、53、54、55、56、57、58 ポインタ
- 71、81、82、83、91、101、108、109、111、112、113、114、115、116、117、122、123、131 文字列矩形
- 72 下線矩形
- 80、121 表矩形
- 92、93、94、95、102、103、104、105、106、107、110 文字矩形

【図1】

【図2】

【図15】

本発明の原理図 システム構成図 重複している外接矩形を示す図



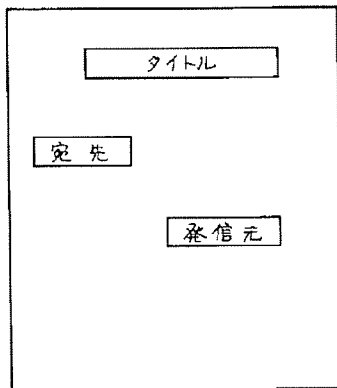
【図3】

文書画像のタイトル抽出処理のフローチャート



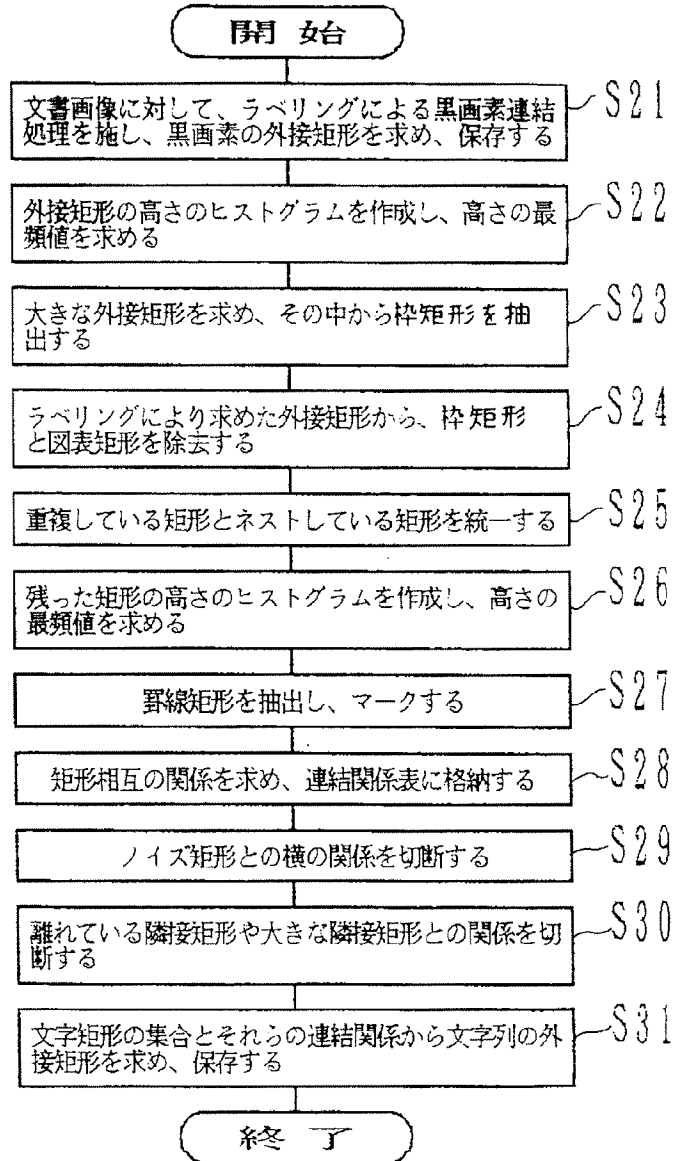
【図4】

タイトルと宛先/発信元の第1の配置を示す図



【図5】

文字列抽出処理のフローチャート



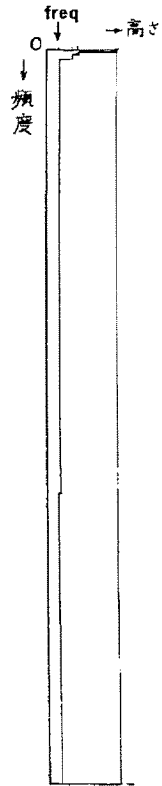
【図6】

ラベリング処理後の外接矩形を示す図



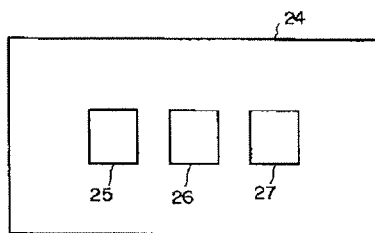
【図8】

高さの最頻値を求めるためのヒストグラムを示す図



【図16】

ネストしている外接矩形を示す図



【図66】

文字矩形と文字数の関係を示す図

文字矩形 高さH 文字数 = $[W / H]$
幅W

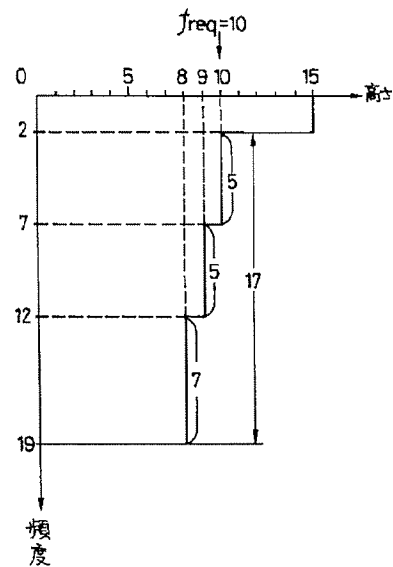
【図9】

矩形高さテーブルを示す図

頻度	最大高さ
2	15
7	10
12	9
19	8

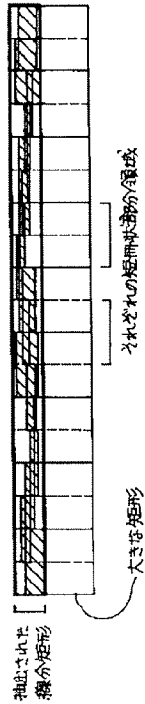
【図10】

矩形高さテーブルの内容に対応するヒストグラムを示す図



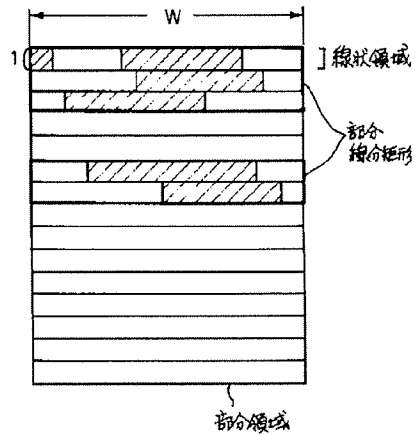
【図11】

大きな矩形から抽出された線分矩形を示す図



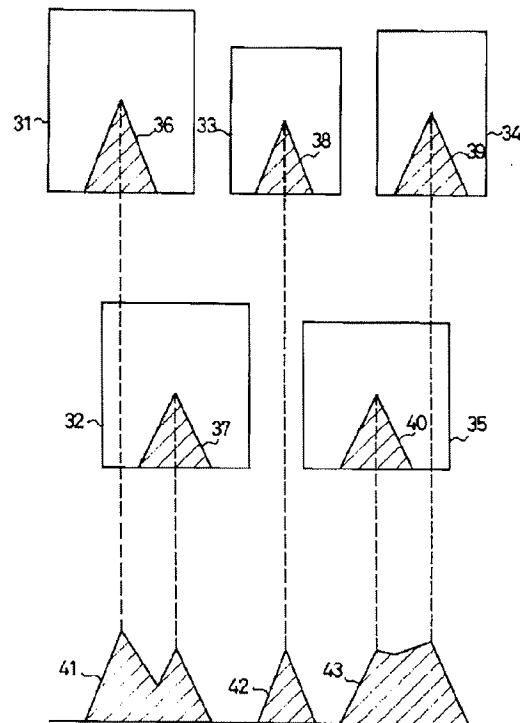
【図12】

部分線分矩形を示す図



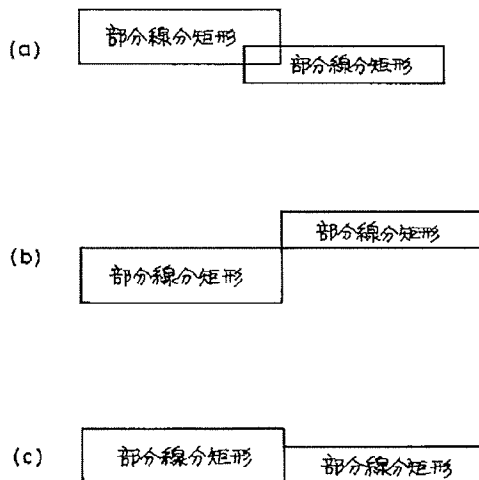
【図17】

二等辺三角形のヒストグラムを示す図



【図13】

連続した部分線分矩形を示す図



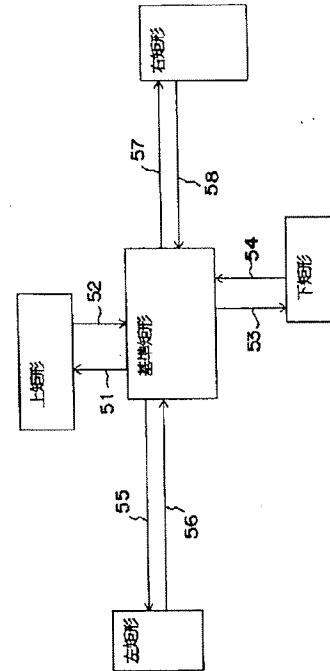
【図18】

重複・ネスト除去後の外接矩形を示す図



【図19】

矩形間の連結関係を示す図



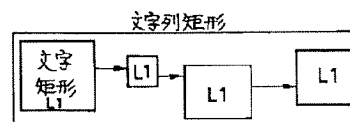
【図20】

連結関係表を示す図

基準矩形のラベル値
上矩形へのポインタ
上矩形からのポインタ
下矩形へのポインタ
下矩形からのポインタ
左矩形へのポインタ
左矩形からのポインタ
右矩形へのポインタ
右矩形からのポインタ

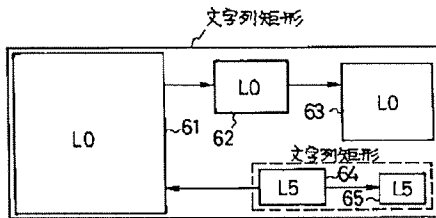
【図21】

文字列矩形を示す図



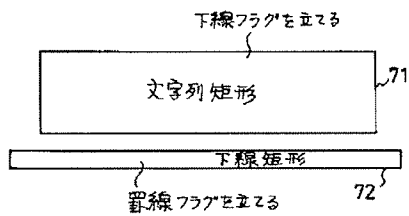
【図22】

文字列矩形の抽出処理を示す図



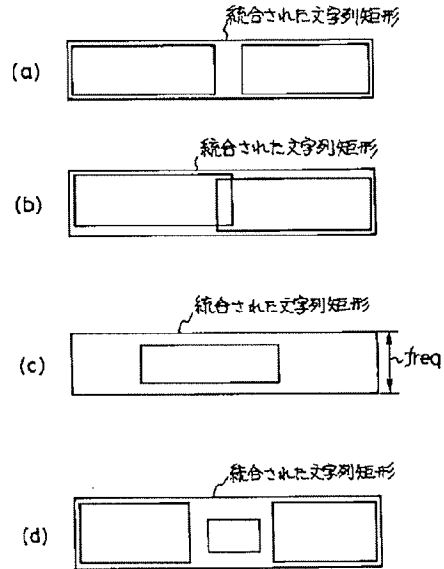
【図29】

下線矩形を示す図



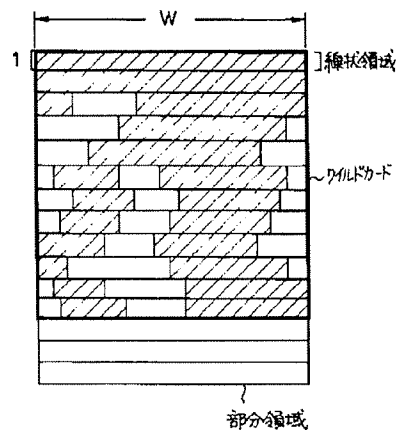
【図26】

文字列矩形の統合処理を示す図



【図33】

ワイルドカードを示す図



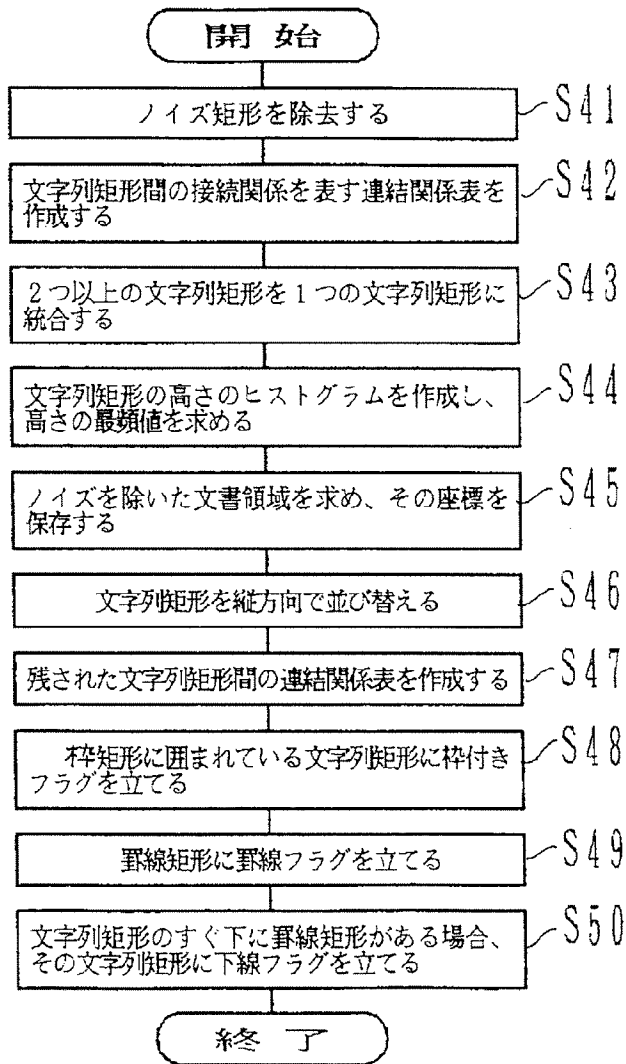
【图23】

抽出された文字列矩形を示す図



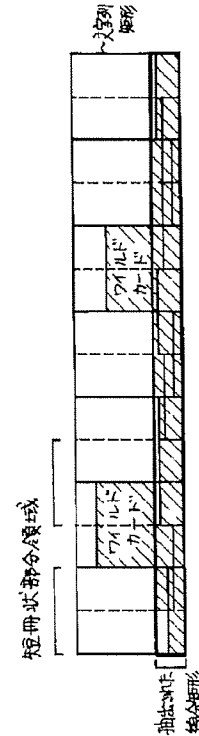
【図24】

文字列矩形加工処理のフローチャート



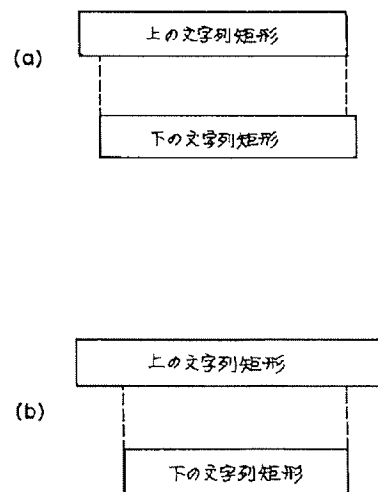
【図32】

フィールドカードがある場合の線分矩形を示す図



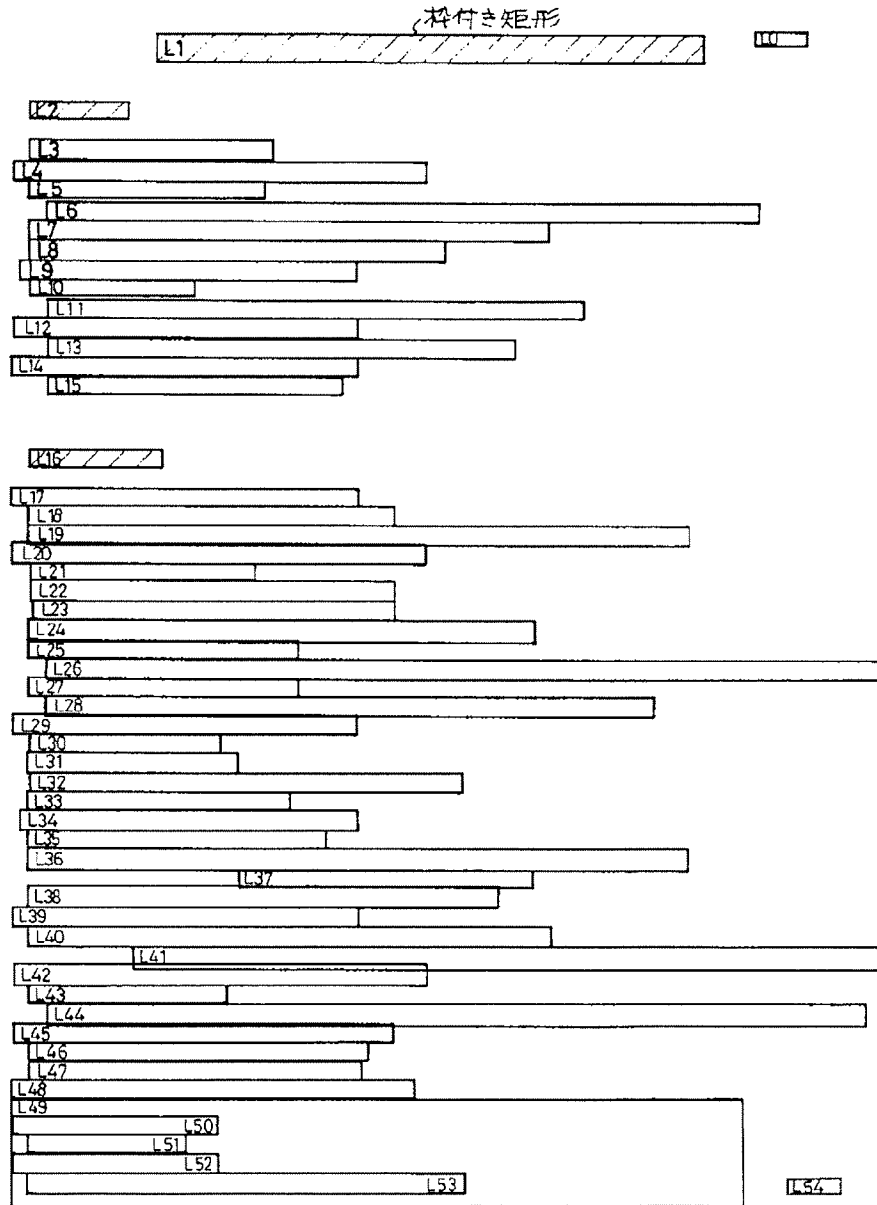
【図41】

オーバーラップしている文字列矩形を示す図



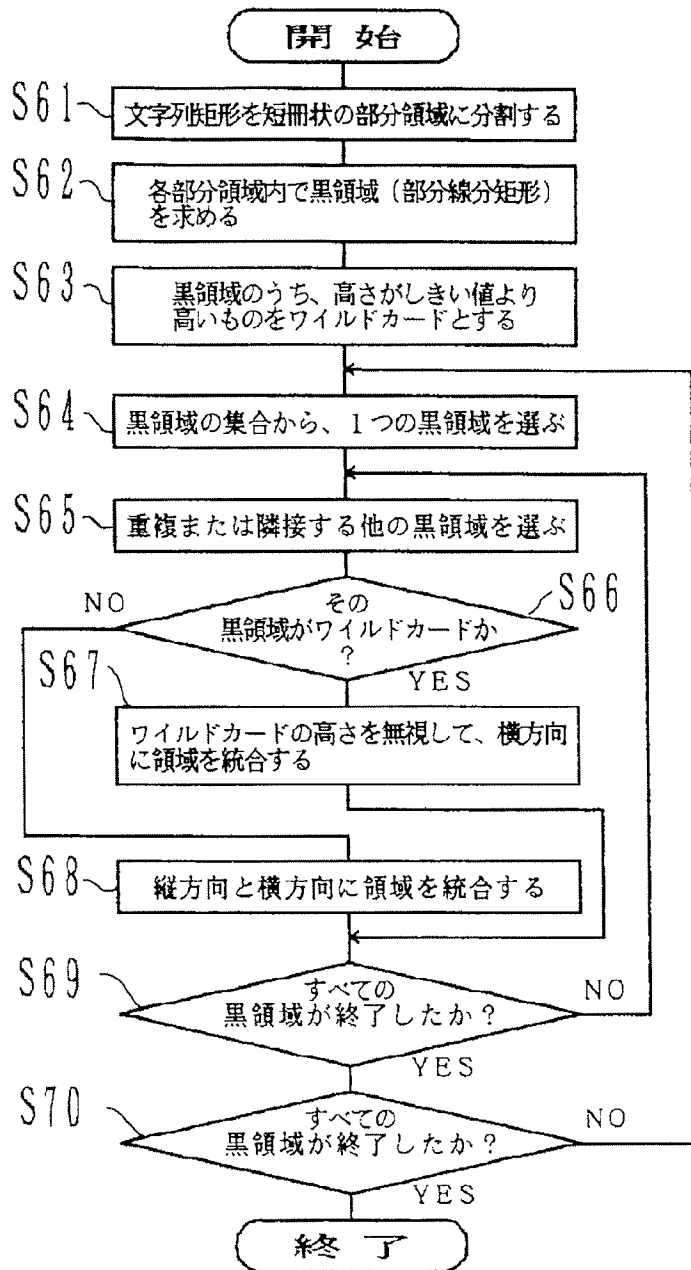
【図30】

枠付き・罫線・下線チェック後の文字列矩形を示す図



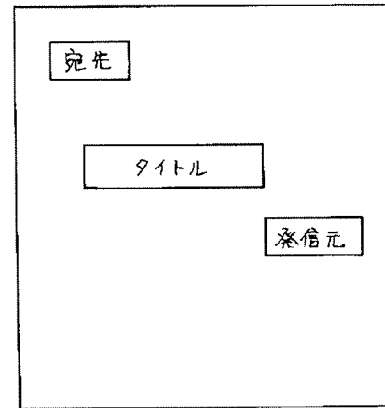
【図31】

線分抽出処理のフローチャート



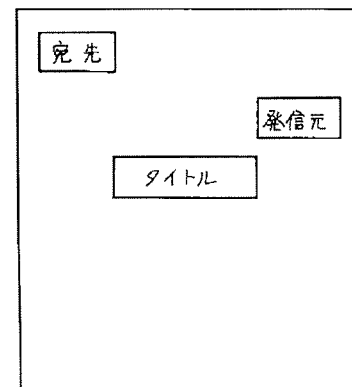
【図45】

タイトルと宛先/発信元の第2の配置を示す図



【図46】

タイトルと宛先/発信元の第3の配置を示す図



【図34】

線分抽出処理のコードを示す図(その1)

先ず、部分矩形の中で文字列矩形の高さ×0.3以上のものをワイルドカード部分矩形としてマークする(use=9)。
それ以外は、普通の矩形としてマークする(use=0)。

lnum=0

部分矩形全部に渡って以下を処理。カレント矩形: i {

```

if( (矩形iのuse が 0) または (矩形iのuseが9) ) {
  xlf =      矩形iの左端座標
  xr  =      矩形iの右端座標
  yup = line__starty = 矩形iの上端座標
  ybl = line__endy   = 矩形iの下端座標

  if(矩形iのuse が 0) {
    standard__st=yup;
    standard__en=ybl;
    standard__h=ybl-yup+1;
    b__use=0; /* 最初の線分がwildcardでなく、standardが設定あり */
    height=ybl-yup+1;
    矩形iのuse = 1;
  }
  else { /* use:9の場合 */
    standard__st=0;
    standard__en=0;
    standard__h=0;
    b__use=9; /* 最初の線分がwildcardで、standardが設定なし */
    height2=ybl-yup+1;
    height=0;
  }
}

```

C1→ α 部分矩形全部に渡って以下を処理。カレント矩形: k {

C2→ β

```

}
/* 全ての線分がwildcard線分だった場合 */
if( (b__use が 9) ) {
  /* 最初の線分の高さを長い線分の高さとする */
  height=height2;
}

```

求めた線分座標(左端: xlf, 右端: xr, 上端: line__starty, 下端: line__endy)を
yokolineのlnum番目に格納

lnumを1つインクリメント

```

}
}

```

【図35】

線分抽出処理のコードを示す図(その2)

```

rxlr = 矩形kの左端座標
rxr  = 矩形kの右端座標
ryup = 矩形kの上端座標
rybl = 矩形kの下端座標
rheight=rybl-ryup+1;

/* standardの値が設定されている */
if( (b_use が 0) ) {
  /* standardありで、右の矩形がwildcard */
  if(矩形kのuseが9) {

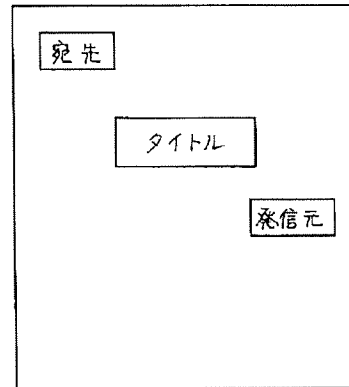
    /* 右隣の矩形が横&縦方向にldot以上の重なり */
    if( ((xr+1)>=rxlf) && (xr < rxr) &&
        ((ybl+1)>=ryup) && ((yup-1)<=rybl) ) {
      xr      = rxr;
    }
  }
  /* standardありで、右の矩形がwildcardではない */
  else if(矩形kのuseが0) {
    /* 右隣の矩形が横&縦方向にldot以上の重なり */
    if( ((xr+1)>=rxlf) && (xr < rxr) &&
        ((ybl+1)>=ryup) && ((yup-1)<=rybl) &&
        (standard_h-TH_HEIGHTDOT <=rheight) &&
        (rheight <=standard_h+TH_HEIGHTDOT) ) {
      矩形kのuse      = 2;
      xr      = rxr;
      yup     = ryup;
      ybl     = rybl;
      hei=rybl-ryup+1;
      if(hei>height) {
        height=hei;
      }
      if(ryup<line__starty)
        line__starty=ryup;
      if(rybl>line__endy)
        line__endy=rybl;

      standard_h=hei;
    }
  }
}

```

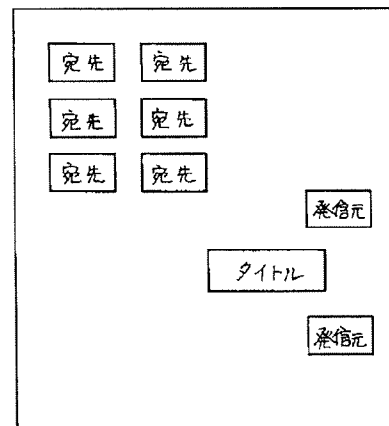
【図47】

タイトルと宛先/発信元の第4の配置を示す図



【図48】

複数の宛先/発信元を示す図



【図36】

線分抽出処理のコードを示す図(その3)

```

/* standardの値が設定されていない */
else if( (b_use == 9) ) {
/* standardなしで、右の矩形がwildcard */
if( (矩形のuseが0) ) {

/* 右隣の矩形が横方向にldot以上の重なり */
if( ((xrl+1)>rxlf) 且 (xr < rxr) ) {
xr = rxr;
}
}
/* standardなしで、右の矩形がwildcardではない */
else if( (矩形のuseが0) ) {

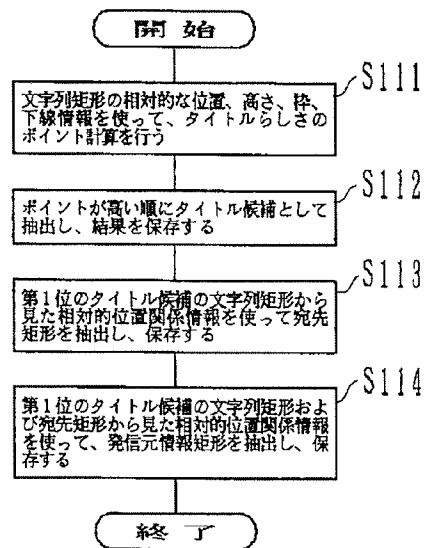
/* 右隣の矩形が横方向にldot以上の重なり */
if( ((xrl+1)>rxlf) 且 (xr < rxr) ) {

b_use=0; /* standard設定済み */
矩形のuse = 2;
xr = rxr;
yup = ryup;
ybl = rybl;
hei=rybl-ryup+1;
if( (hei>height) ) {
height=hei;
}
standard_st=ryup;
standard_en=rybl;
standard_h=hei;
line_starty=ryup;
line_endy=rybl;
}
}
}

```

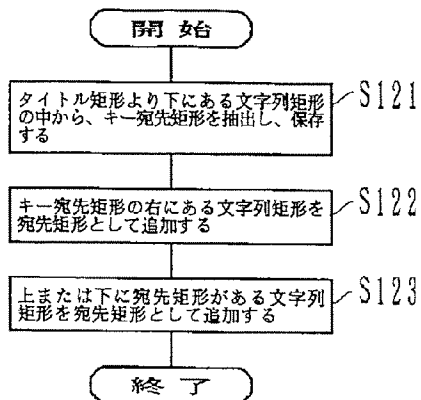
【図40】

タイトル・宛先・発信元抽出処理のフローチャート



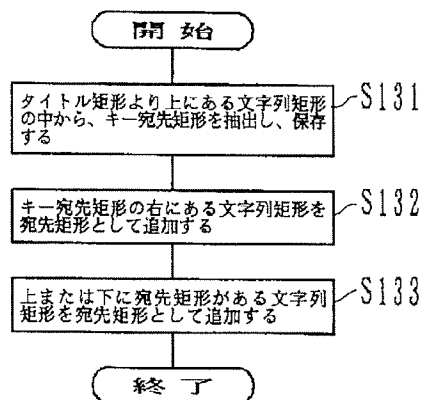
【図42】

第1の宛先抽出処理のフローチャート



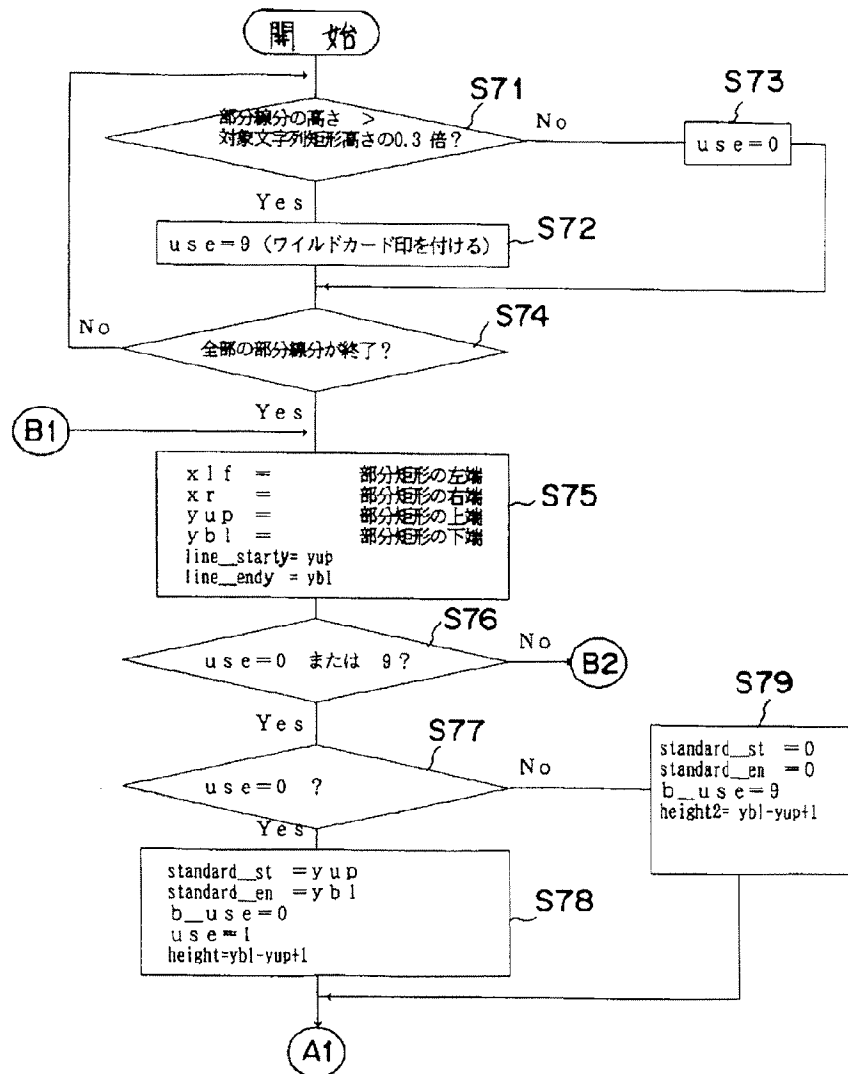
【図43】

第2の宛先抽出処理のフローチャート



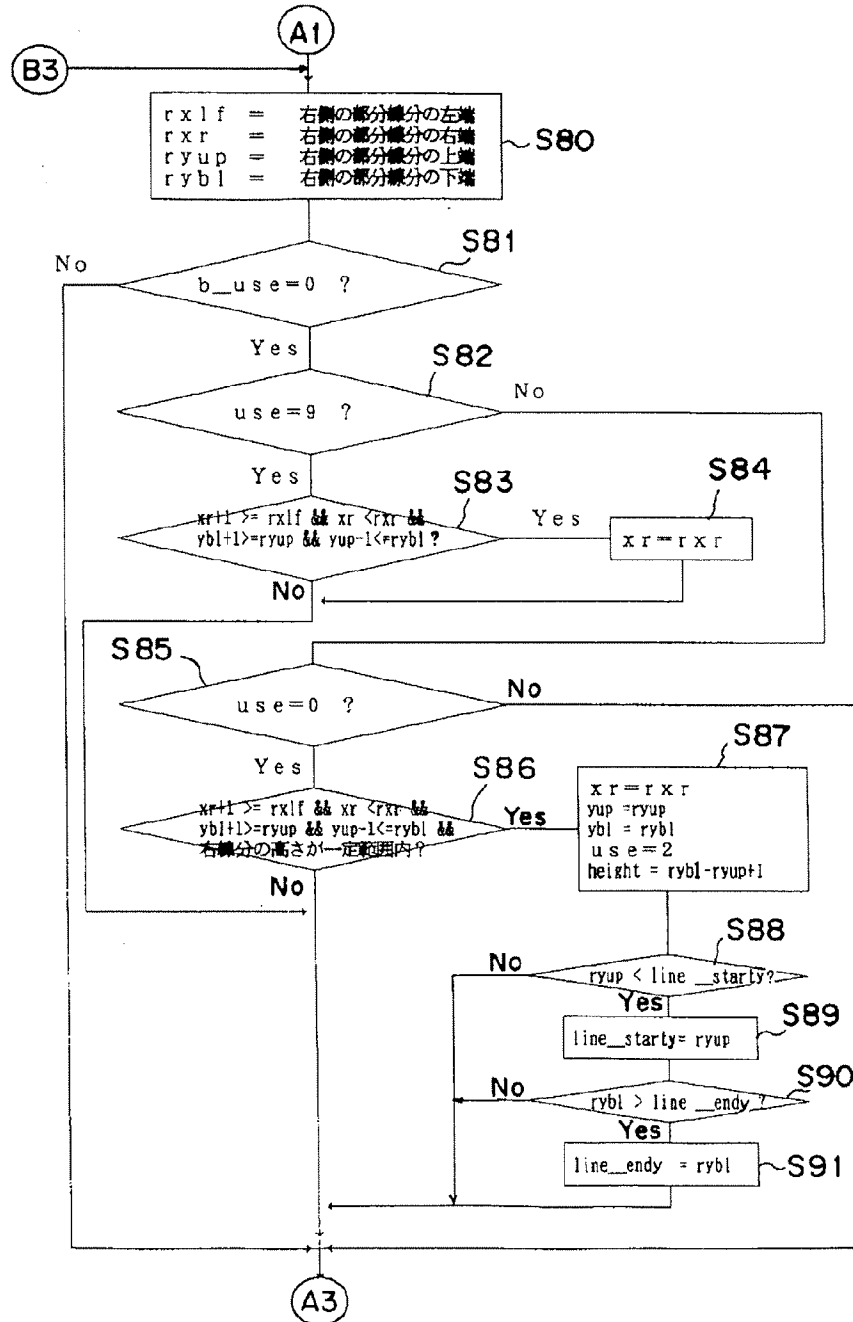
【図37】

線分抽出処理の詳細フローチャート(その1)



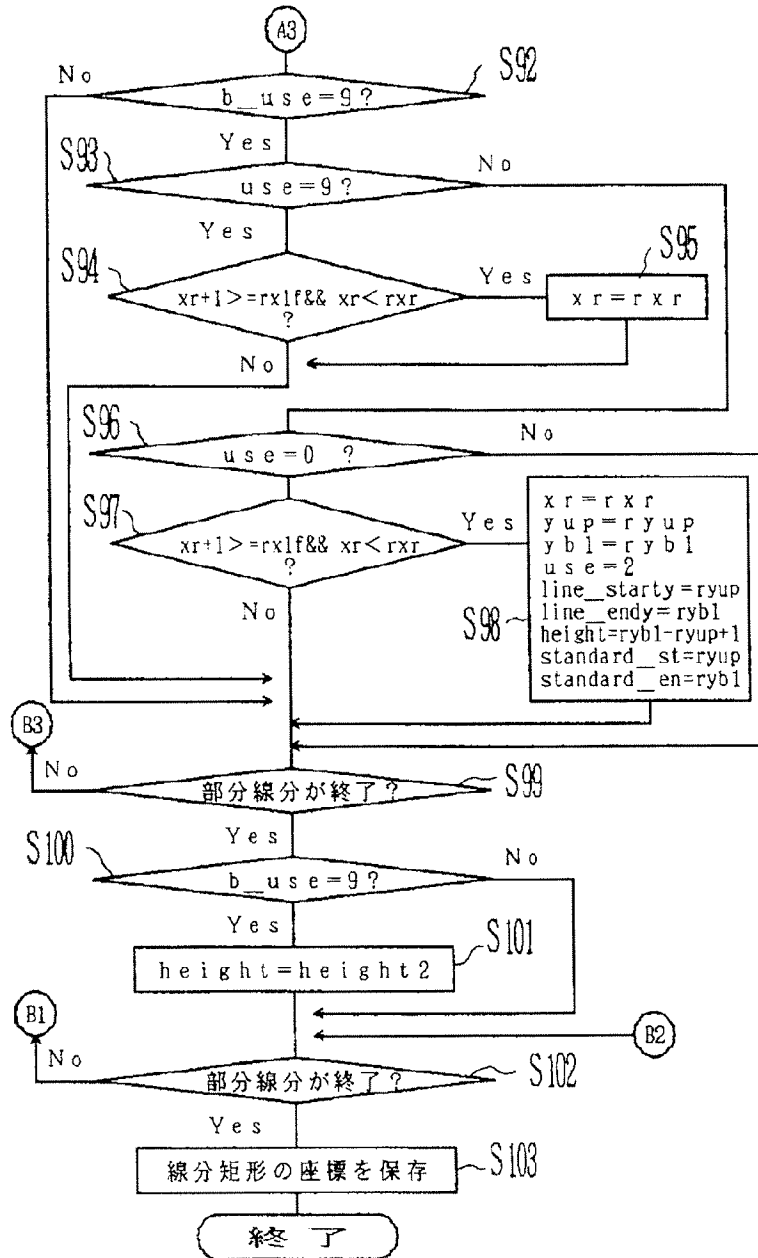
【図38】

線分抽出処理の詳細フローチャート(その2)



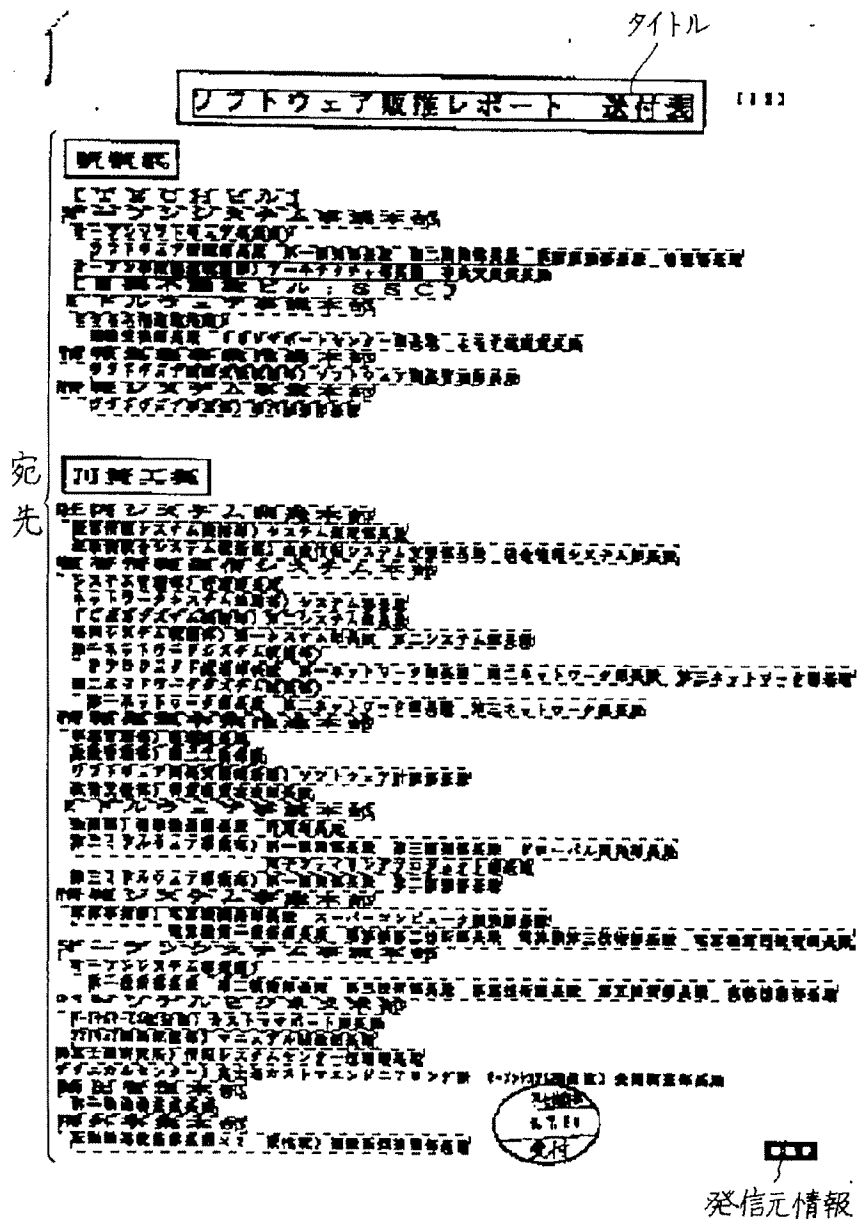
【図39】

線分抽出処理の詳細フローチャート (その3)



【图49】

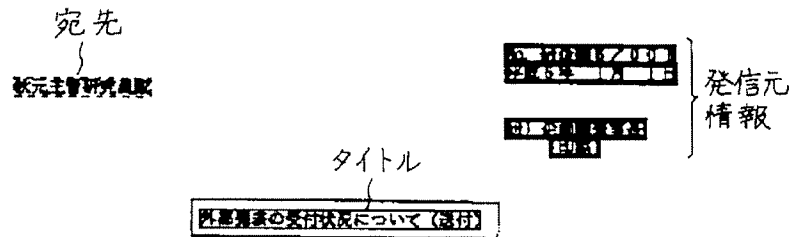
タイトルおよび宛先・発信元情報の抽出結果を示す図



【図50】

タイトルおよび宛先・発信元情報の
他の抽出結果を示す図

technical_news



通知・企画] 要領より下記の資料が発行されましたので送付いたします。活動推進等
にご利用下さい。
なお、「社外発表 受付リスト(平成6年6月度)」は企画調査室にて保管しております。
必要の方は随時までご連絡下さい。

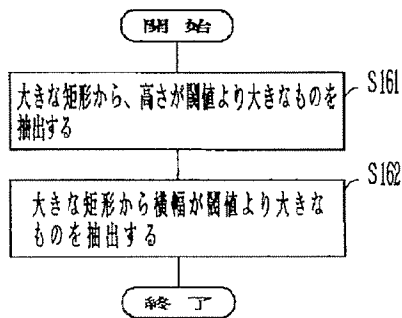
記

添付書類: (1) 「外資系企業発表許可証」受付件数(平成6年6月度) 1紙

以上
[担当: 中村 02-0059]

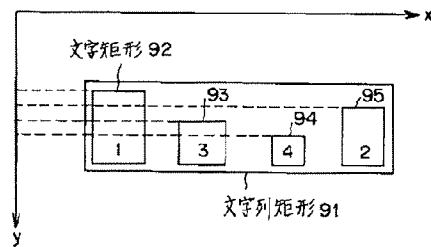
【図55】

表矩形抽出処理のフローチャート



【図58】

文字列矩形内の文字矩形の順位を示す図



【図51】

表形式文書を示す図

出張報告書				回覧		1/18	
表題		マシナリとパターン認識シンポジウム		発行No. : 0123456789-1995		発行元 : ○○○事業部△△△部×××課	
				発行日 : 平成 7年12月31日			
報告者				公開範囲		1: 事業部内 2: 部門内 3: 部内 4: 課内	
従業員番号	氏名	従業員番号	氏名	報告書の性格			
1) 11111	勝山 裕	2)		1: 普通 4: 海外 ①: 速報 2: 急ぎ 5: 重要 2: 詳細 3: 国内 6: その他			
3)		4)					
5)		6)					
7)		8)					
9)		10)					
<p>要約 (350文字) : パターン認識研究の中で、文字認識は、郵便番号読み取り装置のように最も明確なニーズをもった研究対象の一つである。画像をテキストコードに変換する文字認識技術はマルチメディア時代のさまざまな新サービスに利用されようとしている。ここではこれから文字認識を研究されようとする方に、現状のサービス形態、技術レベルについて概観する。文字認識方式にはオフライン型とオンライン型の2種類がある。紙に書かれた文字をスキャナで読み取って認識する方式はオフライン文字認識と呼ばれ、郵便番号読み取り装置、帳票OCR、文書OCR等がある。一方、最近話題を競んだペン入力コンピュータによる文字認識方式はオンライン文字認識と呼ばれ、ペンの筆記過程を時々刻々コンピュータに取り込み、その情報をもとに手書き文字を認識する。</p>							
社内分類	独自分類	通 番	関連出張報告書	関連番号	関連製品名		
(1) 1 2 3 4 (2) 5 6 7 8 (3)	(1) 2 3 4 (2) 4 5 6 (3)	(1) 0 0 0 1 (2) (3) (4) (5)	(1) (2) (3) (4) (5)	(1) A A A 0 1 (2) B B B 0 2 (3) C C C 0 3 (4) D D D 0 4 (5) E E E 0 5	(1) 文書リーダ (2) 帳票OCR (3) (4) (5)		
キーワード (1) OCR (2) 画像処理 (3) 領域抽出 (4) 文書構成要素 (5) 表 (6) (7) (8) (9)			(2) 文字認識 (3) ラベリング (4) 電子ファイリングシステム (5) タイトル (6) 項目 (7) (8) (9)	所属長印			
				受 付 印			
関係配布先名		コード	関係配布先名		コード		
(1) 鈴木課長殿 (2) 山本部長殿 (3) (4) (5) (6) (7) (8) (9) (10)		1 1 1 1 3 3 3 3	(2) 佐藤課長殿 (3) 加藤課長殿 (4) (5) (6) (7) (8) (9) (10)		2 2 2 2 4 4 4 4		
						整理番号	
						合計配布部数 4 部	
複写の発行元確認 ①: 要 2: 不要				連絡先・電話番号 氏名 勝山 裕 ☎777-1111			

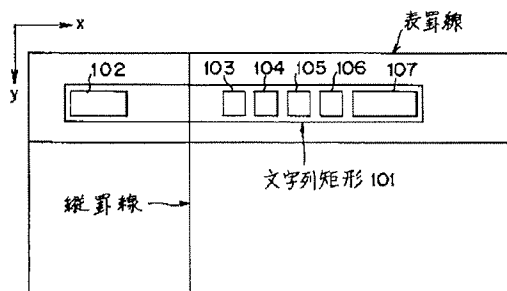
【図52】

表内タイトル抽出処理のフローチャート



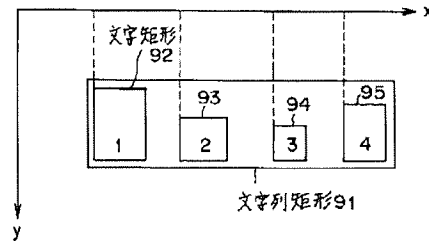
【図60】

縦罫線を含む文字列矩形を示す図



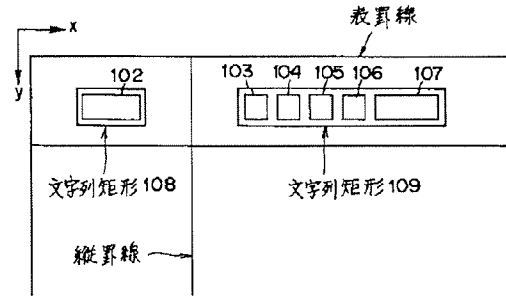
【図59】

ソート後の文字矩形の順位を示す図



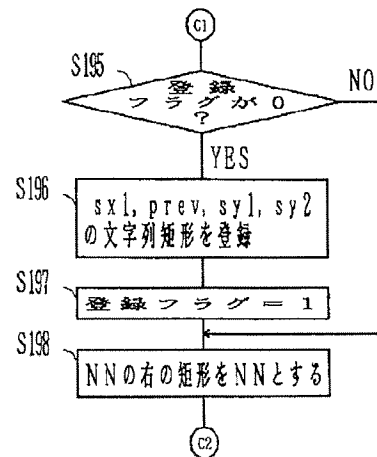
【図61】

分割された文字列矩形を示す図



【図63】

第2の文字列分割処理のフローチャート (その2)



【图53】

表形式文書の画像データを示す図

[illegible]

【図54】

表形式文書のラベリング結果を示す図

表矩形 80

[illegible]

【図56】

表形式文書の文字列矩形を示す図

The diagram shows a form with the following structure:

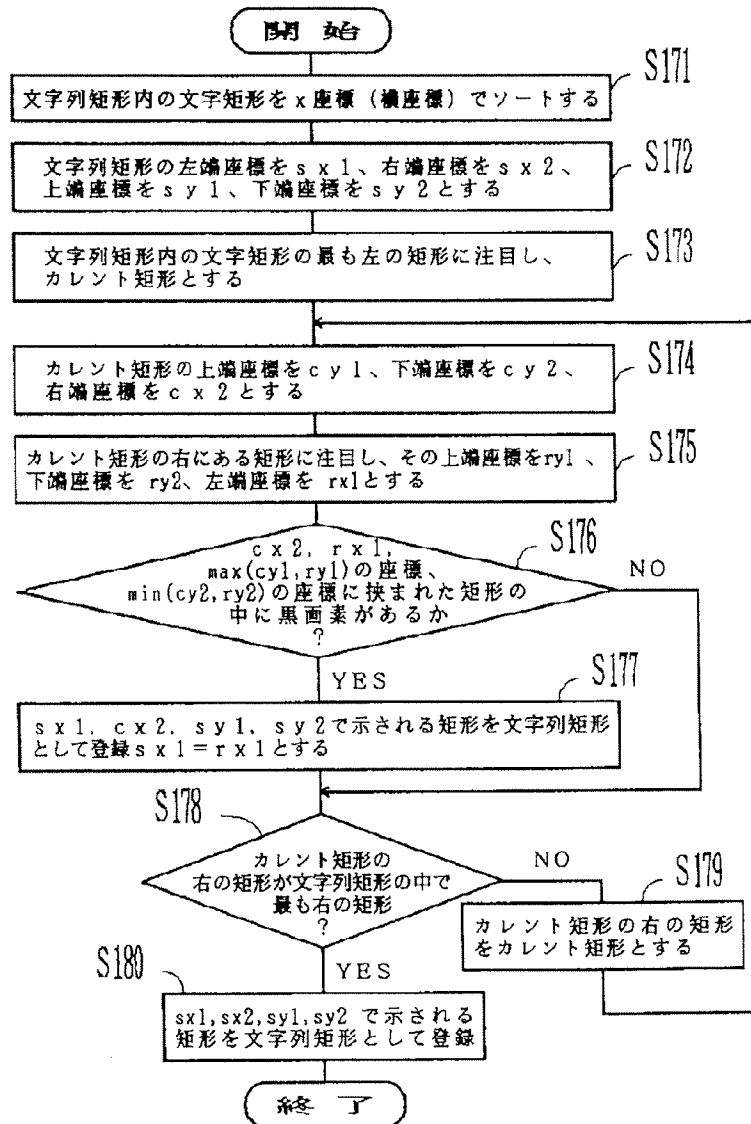
- Header:** Contains a title bar and a date field.
- Table:** A large table with multiple columns and rows. The columns are labeled with various codes and names. The rows contain data, including names and dates.
- Footer:** Contains a summary section and a total row.

Labels in the diagram:

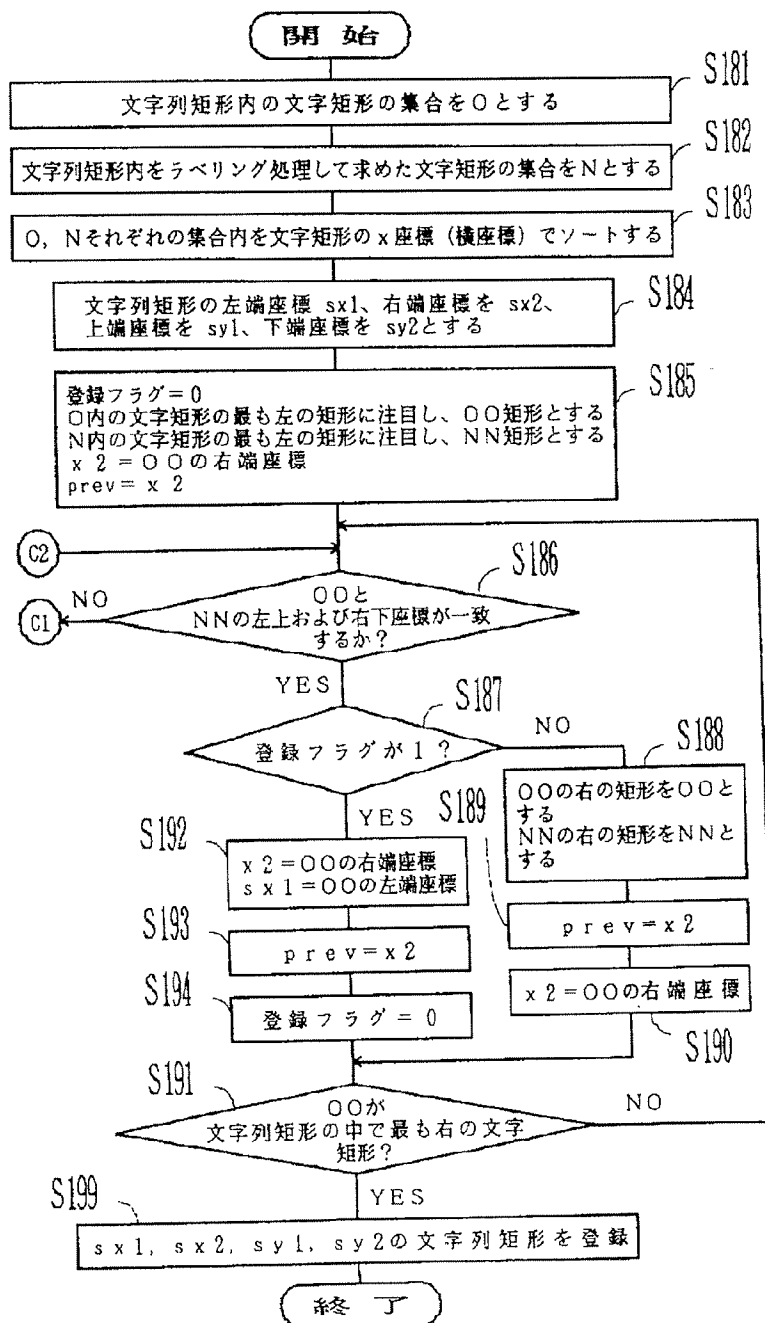
- 81:** Points to the title bar area.
- 82:** Points to the first column of the table.
- 83:** Points to a specific cell in the table.

【図57】

第1の文字列分割処理のフローチャート

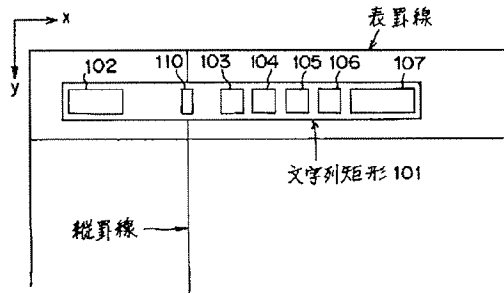


第2の文字列分割処理のフローチャート(その1)



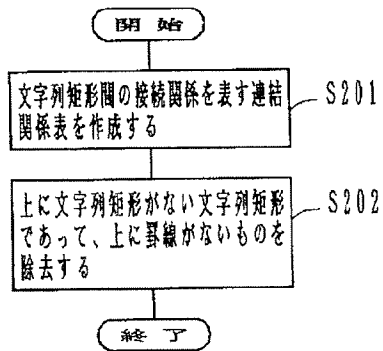
【図64】

文字列矩形内のラベリング結果を示す図



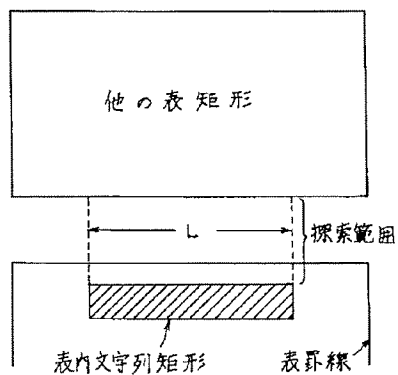
【図69】

上罫線チェック処理のフローチャート



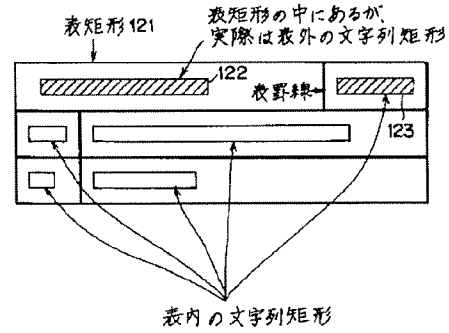
【図72】

第2の探索範囲を示す図



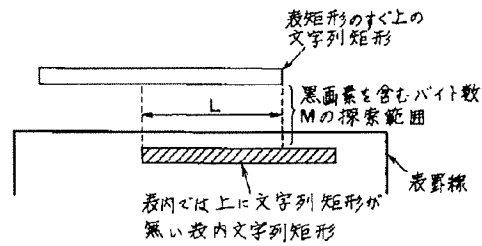
【図67】

表矩形内の表外文字列矩形を示す図



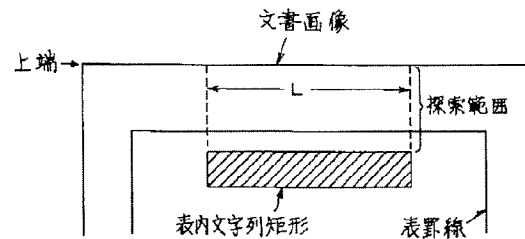
【図71】

第1の探索範囲を示す図



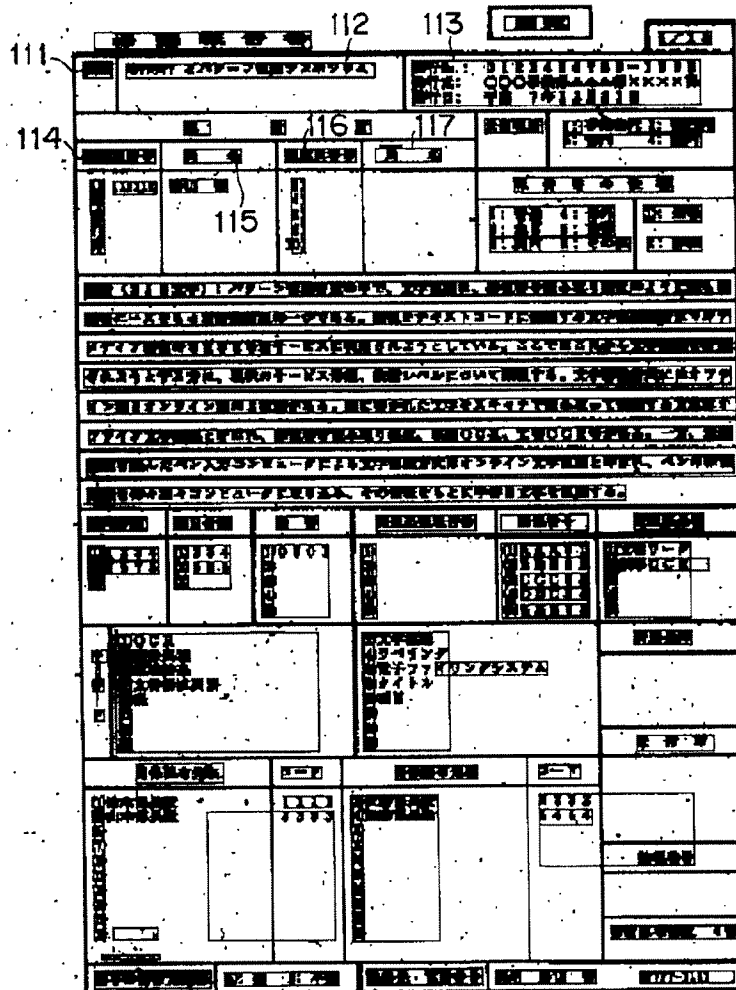
【図73】

第3の探索範囲を示す図



【図65】

分割処理後の文字列矩形を示す図



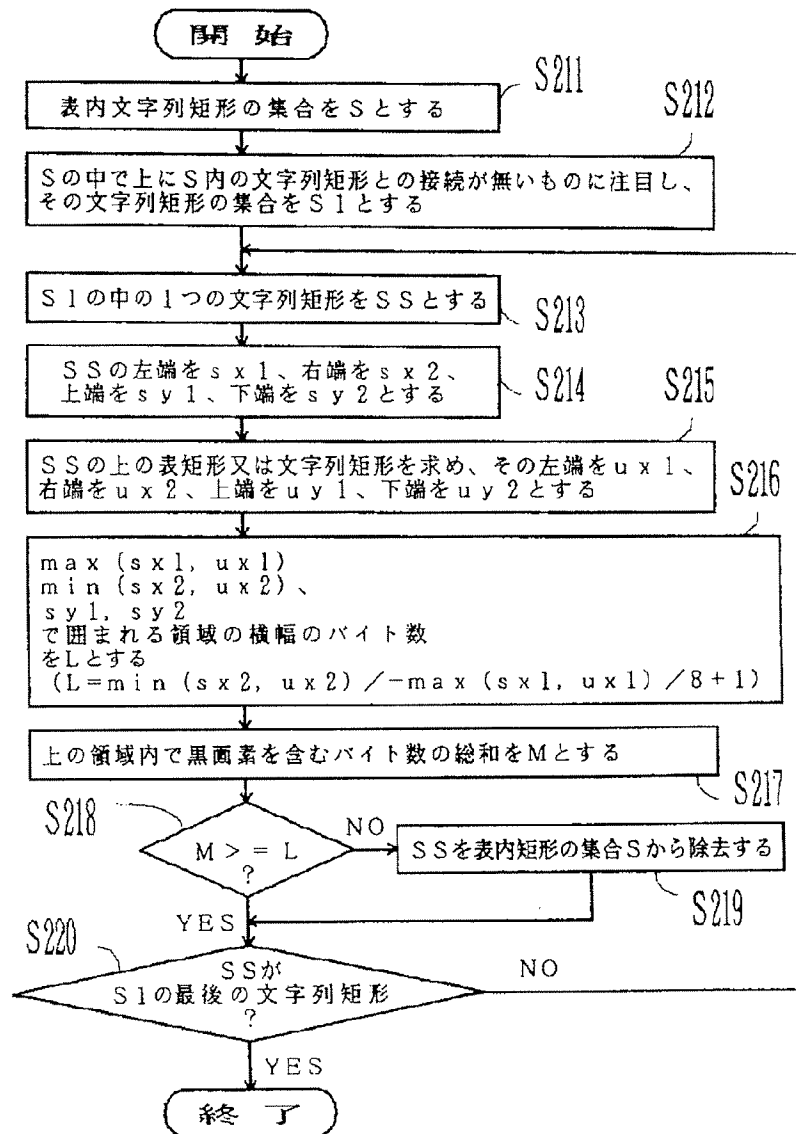
【図68】

表矩形内の文字列矩形形を示す図

[illegible]

【図70】

表外文字列矩形除去処理のフローチャート



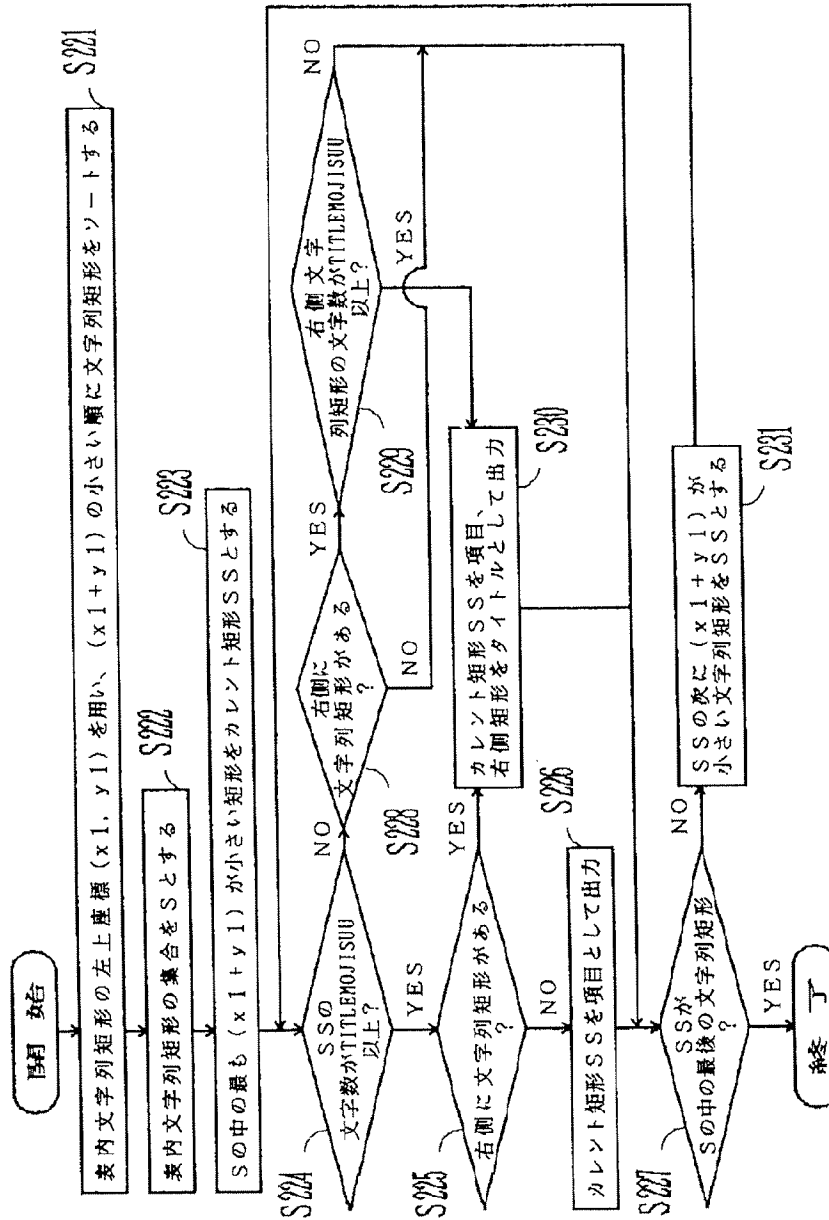
【図74】

表外文字列矩形除去後の文字列矩形を示す図

[illegible]

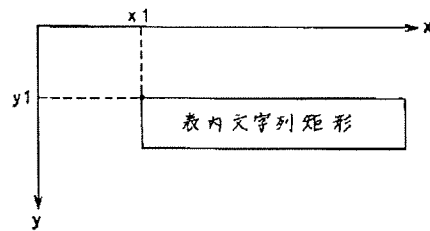
【図75】

タイトル候補出力処理のフローチャート



【図76】

文字列矩形の左上頂点の座標を示す図



【図77】

表内タイトルの抽出結果を示す図

11		12		13		14	
15				16			
17				18			
19				20			
21				22			
23				24			
25				26			
27				28			
29				30			
31				32			
33				34			
35				36			
37				38			
39				40			
41				42			
43				44			
45				46			
47				48			
49				50			
51				52			
53				54			
55				56			
57				58			
59				60			
61				62			
63				64			
65				66			
67				68			
69				70			
71				72			
73				74			
75				76			
77				78			
79				80			
81				82			
83				84			
85				86			
87				88			
89				90			
91				92			
93				94			
95				96			
97				98			
99				100			